Jetter automation

# User Manual

JXM-IO-E02 - I/O-Modul am CAN-Bus

60877286

We automate your success.

Variant: Jetter
Item #: 60877286
Revision 3.29.2
Februar 2015 / Printed in Germany

This document has been compiled by Jetter AG with due diligence, and based on the known state of the art.

In the case of modifications, further developments or enhancements to products shipped in the past, a revised document will be supplied only if required by law, or deemed appropriate by Jetter AG. Jetter AG shall not be liable for errors in form or content, or for missing updates, as well as for damages or disadvantages resulting from such failure.

The logos, brand names, and product names mentioned in this document are trademarks or registered trademarks of Jetter AG, of associated companies or other title owners and must not be used without consent of the respective title owner.

**Address**

How to contact us:

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg
Germany

| | |
|---|---|
| Phone - Switchboard: | +49 7141 2550-0 |
| Phone - Sales: | +49 7141 2550-433 |
| Phone - Technical Hotline: | +49 7141 2550-444 |
| Fax - Sales: | +49 7141 2550-484 |
| E-mail - Sales: | sales@jetter.de |
| E-mail - Technical Hotline: | hotline@jetter.de |


**Assignment to product**

This User Manual is an integral part of JXM-IO-E02:

Type: _____

Serial #: _____

Year of manufacture: _____

Order #: _____

CE

To be entered by the customer:

Inventory #: _____

Place of operation: _____

**Significance of this User Manual**

This document is an integral part of the JXM-IO-E02:

- Keep this document in a way that it is always at hand until the JXM-IO-E02 will be disposed of.
- Pass this document on if the JXM-IO-E02 is sold or loaned/leased out.

In any case you encounter difficulties to clearly understand the contents of this document, please contact Jetter AG.

We would appreciate any suggestions and contributions on your part and would ask you to contact us at the following e-mail address: info@jetter.de. Your feedback will help us produce manuals that are more user-friendly, as well as address your wishes and requirements.

This document contains important information on the following topics:

- Transport
- Mounting
- Installation
- Programming
- Operation
- Maintenance
- Repair

Therefore, you must carefully read, understand and observe this document, and especially the safety instructions.

In the case of missing or inadequate knowledge of this document Jetter AG shall be exempted from any liability. Therefore, the operating company is recommended to obtain the persons' confirmation that they have read and understood this manual in writing.

# Hazard levels

**Introduction**

This topic describes the safety labels and hazard levels used in this manual.

**Safety labels**

Signs using this symbol are to warn you of injuries or even death. Follow the instructions given in the corresponding topic to prevent hazards.

**Hazard levels**

Safety information is classified into the following hazard levels:

| Hazard level | Consequences | Probability |
|---|---|---|
| DANGER | Death/severe injury (irreversible) | The hazard is imminent. |
| WARNING | Death/severe injury (irreversible) | Potential occurrence |
| CAUTION | Slight injury (reversible) | Potential occurrence |
| CAUTION | Material damage | Potential occurrence |

# Table of Contents

Hazard levels ..... 5

**1 Safety instructions 9**

Basic safety instructions ..... 10
Residual dangers and protective measures ..... 12

**2 Product description and design 13**

JXM-IO-E02 - Product description ..... 14
Parts and interfaces ..... 15
Order reference/options ..... 16
Physical dimensions ..... 17

**3 Identifying the module 19**

**3.1 Identification by means of the nameplate ..... 20**
Nameplate ..... 21
**3.2 Identification via CANopen® bus ..... 22**
Electronic Data Sheet (EDS) and software version of the module ..... 23

**4 Mounting and installation 25**

**4.1 Wiring ..... 26**
Wiring principle ..... 27
CAN bus cable - Specification ..... 29
Pin assignment - Overview ..... 31
Connecting the power supply and the 5 V output ..... 33
CAN Interface and Node ID ..... 35
Connecting Digital Inputs and Outputs ..... 37
Connecting Analog Inputs and Outputs ..... 42
**4.2 Installing the JXM-IO-E02 ..... 45**
Installing the JXM-IO-E02 ..... 46

**5 Initial commissioning 51**

Preparatory work for initial commissioning ..... 52
Information on communication with a JXM-IO-E02 ..... 54

**6 CANopen® objects 57**

**6.1 CANopen® object dictionary for JXM-IO-E02 ..... 58**
Objects ranging from index 0x1000 through 0x2000 ..... 60
Object "Digital Outputs" (Index 0x2100) ..... 62
Objekt " Digital Universal-I/O" (Index 0x2101) ..... 64
Object "Tri-state Input" (Index 0x2102) ..... 66
Object "Switch Feed Output" (Index 0x2103) ..... 68
Objects "Analog Input" (Index 0x2200 ... 0x2203) ..... 69
Object "Voltage Sense Analog Input" (Index 0x2210) ..... 71
Object "Feed Currents" (Index 0x2211) ..... 72
Object "Analog Output" (Index 0x2300) ..... 73

Objects "PWM Output" (Index 0x2400 ... 0x2402) ..... 75
Object "H-Bridge" (index 0x2500) ..... 80
Objects "Frequency Input" (Index 0x2600 ... 0x2601) ..... 82
Object "5 V Output" (0x2800) ..... 84
Objekte "OS Update" (Index 0x4554) und EDS (Index 0x4555) ..... 85
Object "System Parameter" (Index 0x4556) ..... 86
Validating and setting system parameters ..... 98
Object "OS Status" (Index 0x4557) ..... 101
Object "Detailed Software Version" (Index 0x4559) ..... 102
Object "Slave CAN Termination" (Index 0x4560) ..... 103
Object "Master CAN Termination" (Index 0x4561) ..... 104
Object "Electronic Name Plate" (Index 0x4565) ..... 105
Object "User EEPROM Access" (Index 0x5000) ..... 106

**6.2 CANopen® PDO Specification ..... 108**
RX PDO allocation on the JXM-IO-E02 ..... 109
TX PDO allocation on the JXM-IO-E02 ..... 110

**7 Protection and diagnostic features - JXM-IO-E02 113**

Standard feed power input (STANDARD FEED) ..... 115
Protected feed power input (PROTECTED FEED) ..... 116
Digital Outputs 1 ... 8 (Standard Outputs) ..... 117
Digital Outputs 9 ... 16 (Safety Outputs) ..... 118
Analog output ..... 119
PWM outputs 1 ... 3 ..... 120
H-bridge ..... 121
Switch feed outputs 1 ... 2 ..... 122
Safety switch (solid-state relay) ..... 123
5 V reference output ..... 124
Generic fault detection ..... 125

**Appendix 127**

**A: Technical data ..... 128**
Technical specifications ..... 129
Physical dimensions ..... 135
Operating parameters - Environment and mechanics ..... 137
Operating parameters - EMC ..... 138

**B: Index ..... 139**

# 1 Safety instructions

**Introduction**

This chapter informs the user of basic safety instructions. It also warns the user of residual dangers, if there are any.

**Contents**


| Topic | Page |
|---|---|
| Basic safety instructions | 10 |
| Residual dangers and protective measures | 12 |

## Basic safety instructions

**Introduction**

This device complies with the valid safety regulations and standards. Jetter AG attaches great importance to the safety of the users.

Of course, the user should adhere to the following regulations:

- Relevant accident prevention regulations
- Accepted safety rules
- EC guidelines and other country-specific regulations

**Intended conditions of use**

Intended conditions of use include operation in accordance with this User Manual.

The JXM-IO-E02 has been designed as a peripheral module for use in commercial vehicles and mobile machines and is intended for connection to an already existing controller. The peripheral module JXM-IO-E02 has been designed for various input and output signals.

The JXM-IO-E02 meets the requirement of the European Automotive EMC Directive for electric/electronic subassemblies.

The JXM-IO-E02 may only be operated within the limits set forth in the technical specifications. The operating voltage of the JXM-IO-E02 is classified as SELV (Safety Extra Low Voltage). Therefore, the JXM-IO-E02 is not subject to the EU Low Voltage Directive.

**Usage other than intended**

The device must not be used in technical systems which to a high degree have to be fail-safe, such as, for example, in ropeways and airplanes.

The JXM-IO-E02 is no safety-related part as per Machinery Directive 2006/42/EC. This device is not qualified for safety-relevant applications and must, therefore, NOT be used to protect persons.

If you intend to operate the device at ambient conditions not being in conformity with the permitted operating conditions, please contact Jetter AG beforehand.

**Personnel qualification**

Depending on the life cycle of the product, the persons involved must possess specific qualifications. The qualifications required to ensure safe handling of the device at different phases of the product life cycle are listed below:

| Product life cycle | Minimum qualification |
|---|---|
| **Transport/storage:** | Trained and instructed personnel with knowledge in handling electrostatically sensitive components |
| **Mounting/installation:** | Specialized personnel with training in electrical/automotive engineering, such as automotive mechatronics fitters |
| **Commissioning/ programming:** | Trained and instructed experts with profound knowledge of, and experience with, automotive/automation technology, such as automotive engineers for mobile machinery |
| **Operation:** | Trained, instructed and assigned personnel with knowledge in operating electronic devices for mobile machinery |

| Product life cycle | Minimum qualification |
|---|---|
| **Decommissioning/ disposal:** | Specialized personnel with training in electrical/automotive engineering, such as automotive mechatronics fitters |

**Modifications and alterations to the module**

**For safety reasons, no modifications and changes to the device and its functions are permitted.**

Any modifications to the device not expressly authorized by Jetter AG will result in a loss of any liability claims to Jetter AG.

**The original parts are specifically designed for the device. Parts and equipment from other manufacturers have not been tested by Jetter AG and are, therefore, not released by Jetter AG.**

The installation of such parts may impair the safety and the proper functioning of the device.

Any liability on the part of Jetter AG for any damages resulting from the use of non-original parts and equipment is excluded.

**Transport**

The JXM-IO-E02 contains electrostatically sensitive components which can be damaged if not handled properly.

To exclude damages to the JXM-IO-E02 during transport it must be shipped in its original packaging or in packaging protecting against electrostatic discharge.

- Use an appropriate outer packaging to protect the JXM-IO-E02 against impact or shock.
- In case of damaged packaging inspect the device for any visible damage. Inform your freight forwarder and Jetter AG.

**Storing**

When storing the JXM-IO-E02 observe the environmental conditions given in the technical specification.

**Repair and maintenance**

The operator is not allowed to repair the device. The device does not contain any parts that could be repaired by the operator.

If the device needs repairing, please send it to Jetter AG.

**Disposal**

When disposing of devices, the local environmental regulations must be complied with.

## Residual dangers and protective measures

**Residual dangers**

Consider the residual dangers mentioned in this chapter when assessing the risks associated with your machine.

**DANGER**

**Hazard in explosive gas atmosphere!**

This device can become a source of ignition in potentially explosive atmospheres.

➢ **Do not use this device in potentially explosive atmospheres.**

**WARNING**

**Hot surface hazard!**

The JXM-IO-E02 can heat up during operation. During operation the surface temperature of this device will become hot enough (> 60 °C) to cause burns.

➢ **Take protective measures to prevent inadvertent contact with the device, e.g. install protective covers.**

➢ Allow the device to cool down for some time before you start working on it, e.g. to carry out maintenance jobs.

**CAUTION**

**Possible occurrence of malfunctions!**

CAN wires which have not been twisted may increase susceptibility to noise. This may disturb communications with the device which, in turn, may cause malfunctions.

➢ Make sure that twisted pair cables are used for connecting the CAN interfaces.

# 2 Product description and design

**Introduction**

This chapter covers the design of the device, as well as how the order reference is made up including all options.

**Contents**


| Topic | Page |
|---|---|
| JXM-IO-E02 - Product description | 14 |
| Parts and interfaces | 15 |
| Order reference/options | 16 |
| Physical dimensions | 17 |

# JXM-IO-E02 - Product description

**The JXM-IO-E02 module**

The module JXM-IO-E02 is an I/O node and has especially been designed for use in the harsh environment of commercial vehicles and mobile machines.

**Product features**

The features of this product are listed below:

- CANopen® node with 1 interface to CAN-2.0B
- 16 digital active-high inputs
- 8 digital active-high outputs supplying up to 2.5 A
- 8 digital active-high outputs supplying up to 5.0 A
- 2 switch feed outputs, 2.5 A max.
- 5 digital inputs which can be configured as active-high or active-low inputs
- 1 analog output (resolution: 10 bits)
- 4 analog inputs (voltage, current, resolution: 10 bits)
- 2 frequency inputs (5 Hz ... 20 kHz, resolution: 62.5 ns)
- 3 PWM outputs, 2.5 A max. (resolution: 10 bits)
- 1 H-bridge, 2.5 A max.
- 2 tri-state inputs for setting the node ID

**Compatibility - Overview**

The following table lists the device/PCB revision numbers and the compatible OS versions:

| Device revision | PCB revision | OS version |
|---|---|---|
| 06.xx and older | 02.xx and older | Unknown (no support) |
| All revisions 07.xx | 03.xx | V 3.16.0.00 and V 3.29.0.00 |
| All revisions 10.xx | 05.xx | V 3.16.0.00 and V 3.29.0.00 |
| All revisions 12.xx | 07.xx | V 3.29.0.00 or higher |

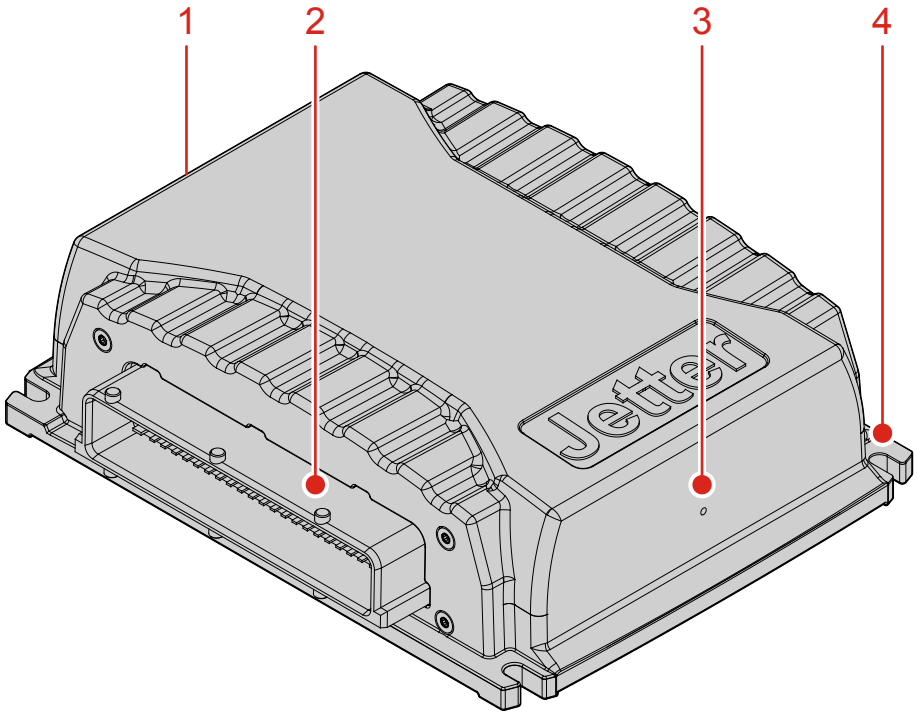
# Parts and interfaces

**Introduction**

This chapter describes the parts and interfaces of the JXM-IO-E02.

**Parts and interfaces**

The JXM-IO-E02 features the following parts and interfaces:




| Number | Element | Function |
|---|---|---|
| **1** | Nameplate | For identifying the JXM-IO-E02 |
| **2** | Female connector | For connecting external components and the controller |
| **3** | Pressure compensation membrane | Compensation of inside and outside air pressure |
| **4** | Fastening lugs | For screwing down the JXM-IO-E02 |

## Order reference/options

**Order reference**

The JXM-IO-E02 is available in different configurations. To order a specific configuration from Jetter AG, please specify the corresponding item number.

| Item no. | Order reference | Name |
|---|---|---|
| 10000818 | JXM-IO-E02-G06-K00 | Standard I/O node |

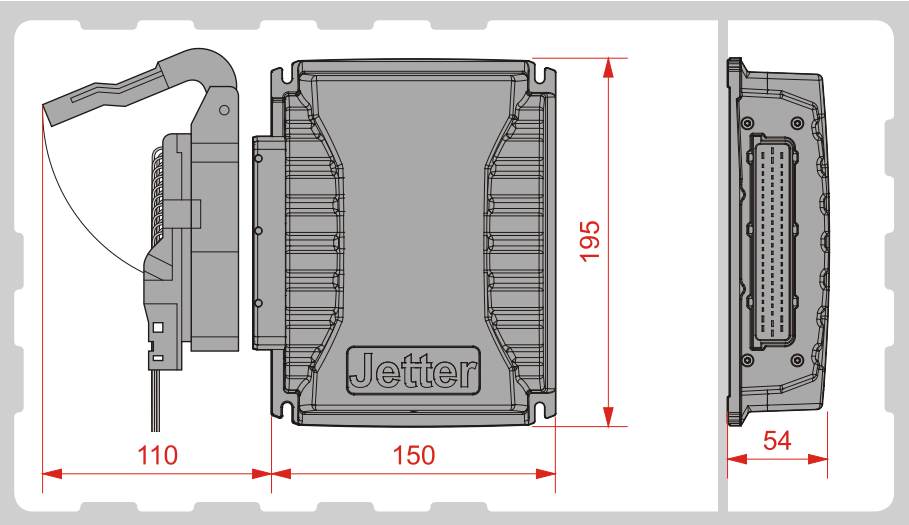
# Physical dimensions

**Introduction**

This chapter details the physical dimensions of the JXM-IO-E02 and the conditions for its installation.

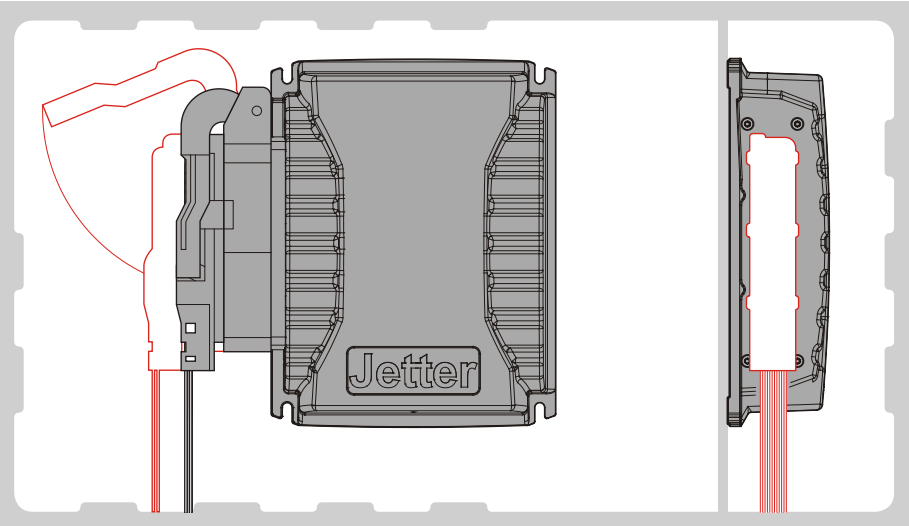
**Physical dimensions**

The illustration shows the physical dimensions of the JXM-IO-E02.




**Space required for installation and service**

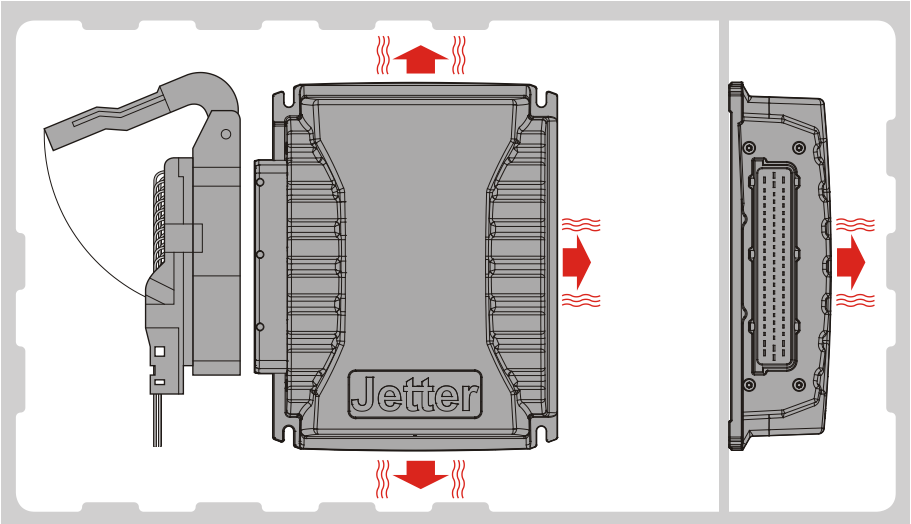
The diagram shows the space required for the JXM-IO-E02.




Ensure there is enough space around the connector for servicing requirements. It should be possible to disconnect the connector at any time.

**Space required to protect against overheating**

The diagram indicates the safety distances to protect against overheating.




Please note:

- The JXM-IO-E02 increases the temperature of the environment as a result of heat emission under load.
- The JXM-IO-E02 operates without interruption at an ambient temperature of up to +85 °C.

Consider the heat emission from the device, in particular when installing it in a critical environment:

- In the vicinity of the fuel tank
- In the vicinity of the fuel pipe
- In the vicinity of flammable vehicle components
- In the vicinity of thermally malleable vehicle components

# 3 Identifying the module

**Purpose of this chapter**

This chapter supports you in retrieving the following information from the JXM-IO-E02:

- Hardware revision
- Retrieving Electronic Data Sheet (EDS) information. Numerous production-relevant data are permanently stored in the EDS.
- Determining the OS version of software components

**Prerequisites**

To be able to identify the JXM-IO-E02 the following prerequisites must be fulfilled:

- The module is connected to a controller and a PC via CANopen® bus.
- Communication with the module takes place via CANopen®.

**Information for hotline requests**

If you wish to contact the hotline of Jetter AG in case of a problem, please have the following information on the JXM-IO-E02 ready:

- Serial number
- Software version
- Hardware revision

**Contents**


| Topic | Page |
|---|---|
| Identification by means of the nameplate | 20 |
| Identification via CANopen® bus | 22 |

# 3.1 Identification by means of the nameplate

**Introduction**

The nameplate is attached to the housing of the JXM-IO-E02 and contains details, such as hardware revision number and serial number. If you wish to contact the hotline of Jetter AG in case of a problem, please have this information ready.

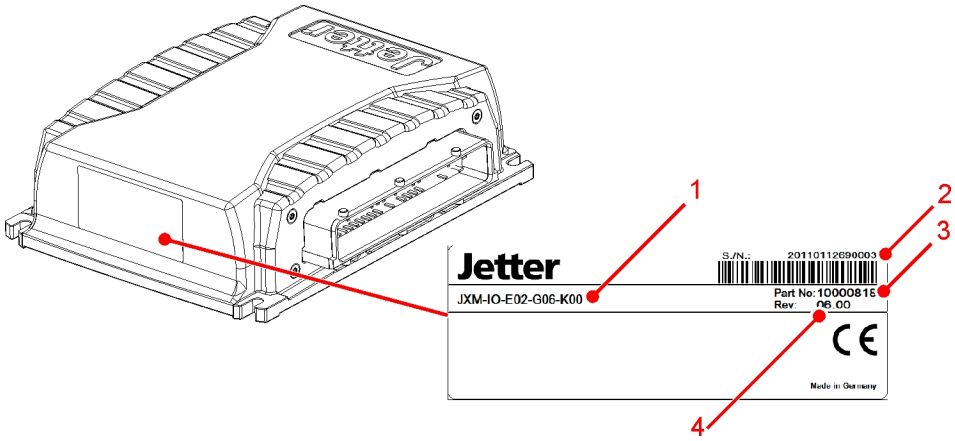
**Contents**


| Topic | Page |
|---|---|
| Nameplate | 21 |

# Nameplate

**Nameplate**

The nameplate of a JXM-IO-E02 contains the following information:




| Number | Description |
|---|---|
| **1** | Module type |
| **2** | Serial number |
| **3** | Item number |
| **4** | Hardware revision |

# 3.2 Identification via CANopen® bus

**Introduction**

Each module features an Electronic Data Sheet (EDS). Numerous production-relevant data are permanently stored in the EDS. EDS data can be retrieved via CANopen® bus.

**Contents**


| Topic | Page |
|---|---|
| Electronic Data Sheet (EDS) and software version of the module | 23 |

## Electronic Data Sheet (EDS) and software version of the module

**Communication with the JXM-IO-E02**

Communication with the JXM-IO-E02 takes place via CAN bus. The CANopen® standard is used as protocol. CANopen® is an open standard for networking and communication in the automobile sector.

The CANopen® protocol has been further developed by the CiA e.V. (CAN in Automation) and works on the physical layer with CAN Highspeed in accordance with ISO 11898.

**Electronic Data Sheet (EDS)**

The Electronic Data Sheet (EDS) provides information clearly identifying the JXM-IO-E02. Data contained in the EDS are production-specific and are relevant for support purposes. The object **Electronic Data Sheet** (0x4555) lets you retrieve EDS information.

**Software version of the JXM-IO-E02**

The object "Detailed Software Version" (0x4559) lets you read out the version of the software running in the JXM-IO-E02. This read-only object returns the same software version as object **0x100A**, but in a 32-bit unsigned integer format which is compatible with the standard IP-type version numbers used at Jetter AG.

Example:

The 32-bit word 0x01070001 translates to a software version of 1.07.00.01.

**Useful documents**

The CANopen® specifications can be obtained from the **CiA e.V. http://www.can-cia.org** homepage. The key specification documents are:

- CiA DS 301 - This document is also known as the communication profile and describes the fundamental services and protocols used under CANopen®.
- CiA DS 302 - Framework for programmable devices (CANopen® Manager, SDO Manager)
- CiA DR 303 - Information on cables and connectors
- CiA DS 4xx - These documents describe the behavior of a number of device classes in, what are known as, device profiles.

**Related topics**

- **Electronic Data Sheet Object** (see page 85)
- **Detailed Software Version Object** (see page 102)

# 4 Mounting and installation

**Purpose of this chapter**

This chapter is to support you in mounting and installing the JXM-IO-E02 in the vehicle and covers the following topics:

- Planning the wiring of a JXM-IO-E02
- Connecting sensors and actuators to the JXM-IO-E02
- Installation
- CANopen® Bus - Engineering

**Contents**


| Topic | Page |
|---|---|
| Wiring | 26 |
| Installing the JXM-IO-E02 | 45 |

# 4.1 Wiring

**Purpose of this chapter**

This chapter describes how to wire the JXM-IO-E02 and covers the following topics:

- Wiring principle
- Pin assignment
- Example of wiring
- Technical specifications

**Contents**


| Topic | Page |
|---|---|
| Wiring principle | 27 |
| CAN bus cable - Specification | 29 |
| Pin assignment - Overview | 31 |
| Connecting the power supply and the 5 V output | 33 |
| CAN Interface and Node ID | 35 |
| Connecting Digital Inputs and Outputs | 37 |
| Connecting Analog Inputs and Outputs | 42 |

# Wiring principle

**Introduction**

This chapter covers the wiring principle of the JXM-IO-E02.

**Wiring principle**

The JXM-IO-E02 is connected through a wiring harness with external components, such as:

- Power supply
- Controller
- Peripheral module
- Sensors
- Actuators
- Indicator lights

The wiring harness connects to the JXM-IO-E02 with a male connector. This connector is not included in the scope of delivery. This connector is available as accessory.

**Connector specification**

The connector specification is listed below:

| Connector specification | |
|---|---|
| Manufacturer/Model | Tyco AMP |
| Item number | 963484 |
| Design | 70-pin |
| Coding | A 1 |

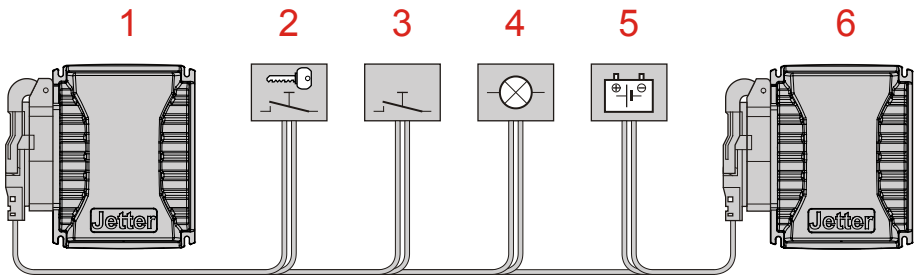
If the connector is used in a humid environment, we recommend to use single wire seals (sealing plugs).

| Single wire seals (large contacts) | |
|---|---|
| Manufacturer | AMP Deutschland GmbH |
| Item number | 828922-1 |

| Single wire seals (small contacts) | |
|---|---|
| Manufacturer | TE Connectivity |
| Item number | 963531-1 |

**Example**

The illustration shows an example of a wiring layout.

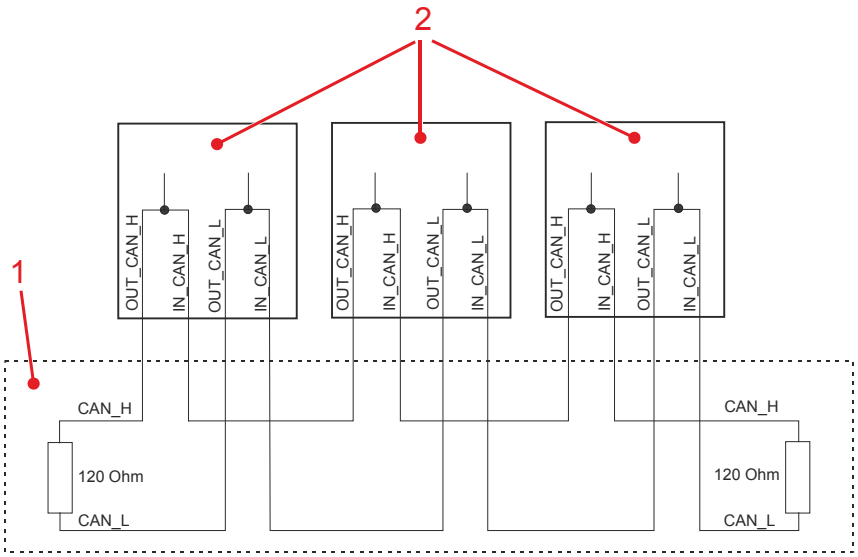



| Number | Description |
|---|---|
| **1** | Module JXM-IO-E02 |
| **2** | Ignition lock |
| **3** | Door contact switch |
| **4** | Indicator light |
| **5** | Battery |
| **6** | Controller JCM-350-E03 |

# CAN bus cable - Specification

**Diagram of the CAN bus wiring**

Wire up Jetter AG CANopen® devices in accordance with the following diagram.




| Number | Description |
|---|---|
| **1** | CAN bus |
| **2** | Jetter AG CANopen® devices |

There is an option to enable a resistor in the device as a bus termination resistor of 120 Ohm.

The stub length with this type of wiring is almost zero.

The CAN_L and CAN_H cables must be twisted together and shielded.

**Specification - CAN bus cable**

| Parameter | Description |
|---|---|
| Core cross-sectional area | 1000 kBaud: 0.25 … 0.34 $mm^2$<br>500 kBaud: 0.34 … 0.50 $mm^2$<br>250 kBaud: 0.34 … 0.60 $mm^2$<br>125 kBaud: 0.50 … 0.60 $mm^2$ |
| Cable capacitance | 60 pF/m max. |
| Resistivity | 1000 kBaud: 70 Ω/km max.<br>500 kBaud: 60 Ω/km max.<br>250 kBaud: 60 Ω/km max.<br>125 kBaud: 60 Ω/km max. |
| Number of cores | 2 |
| Shielding | Complete shielding, no paired shielding |
| Twisting | Core pairs CAN_L and CAN_H are twisted |

**Cable lengths**

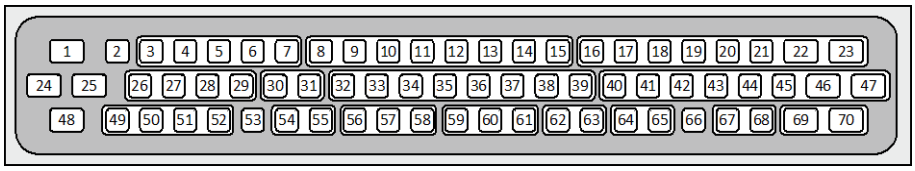
The maximum permitted cable length depends on the baud rate used and the number of CANopen® devices connected.

| Baud rate | Cable length | Stub length | Total stub length |
|---|---|---|---|
| 1000 kBaud | 25 m max. | 0.3 m max. | 1.5 m |
| 500 kBaud | 100 m max. | 5 m max. | 30 m |
| 250 kBaud | 250 m max. | 10 m max. | 60 m |
| 125 kBaud | 500 m max. | 20 m max. | 120 m |

# Pin assignment - Overview

**Connector pinout**

The diagram below shows the pinout of the connector (as viewed from the front):




| Pin | Description | Pin | Description | Pin | Description |
|---|---|---|---|---|---|
| **1** | PROTECTED FEED (+12 VDC or +24 VDC) | **25** | Ground: Analog output | **49** | Analog input # 1 |
| **2** | Ignition (+) (IGNITION FEED) | **26** | Ground: Analog input 1 | **50** | Analog input # 2 |
| **3** | Digital input 1 | **27** | Ground: Analog input 2 | **51** | Analog input # 3 |
| **4** | Digital input 2 | **28** | Ground: Analog input 3 | **52** | Analog input # 4 |
| **5** | Digital input 3 | **29** | Ground: Analog input 4 | **53** | Analog output |
| **6** | Digital input 4 | **30** | Switch feed output 1 | **54** | Frequency input # 1 |
| **7** | Digital input 5 | **31** | Switch feed output 2 | **55** | Frequency input # 2 |
| **8** | Universal I/O: IN 6/OUT 1 | **32** | Ground return: IN 6/OUT 1 | **56** | PWM output 1 |
| **9** | Universal I/O: IN 7/OUT 2 | **33** | Ground return: IN 7/OUT 2 | **57** | PWM output 2 |
| **10** | Universal I/O: IN 8/OUT 3 | **34** | Ground return: IN 8/OUT 3 | **58** | PWM output 3 |
| **11** | Universal I/O: IN 9/OUT 4 | **35** | Ground return: IN 9/OUT 4 | **59** | Ground return: PWM output 1 |
| **12** | Universal I/O: IN 10/OUT 5 | **36** | Ground return: IN 10/OUT 5 | **60** | Ground return: PWM output 2 |
| **13** | Universal I/O: IN 11/OUT 6 | **37** | Ground return: IN 11/OUT 6 | **61** | Ground return: PWM output 3 |
| **14** | Universal I/O: IN 12/OUT 7 | **38** | Ground return: IN 12/OUT 7 | **62** | OUT_CAN_L |
| **15** | Universal I/O: IN 13/OUT 8 | **39** | Ground return: IN 13/OUT 8 | **63** | OUT_CAN_H |
| **16** | Universal I/O: IN 14/PROTECTED OUT 9 | **40** | Ground return: IN 14/PROTECTED OUT 9 | **64** | IN_CAN_L |
| **17** | Universal I/O: IN 15/PROTECTED OUT 10 | **41** | Ground return: IN 15/PROTECTED OUT 10 | **65** | IN_CAN_H |
| **18** | Universal I/O: IN 16/PROTECTED OUT 11 | **42** | Ground return: IN 16/PROTECTED OUT 11 | **66** | Regulated output +5 VDC |
| **19** | Universal I/O: IN 17/PROTECTED OUT 12 | **43** | Ground return: IN 17/PROTECTED OUT 12 | **67** | Node ID (tri-state input # 1) |
| **20** | Universal I/O: IN 18/PROTECTED OUT 13 | **44** | Ground return: IN 18/PROTECTED OUT 13 | **68** | Node ID (tri-state input # 2) |
| **21** | Universal I/O: IN 19/PROTECTED OUT 14 | **45** | Ground return: IN 19/PROTECTED OUT 14 | **69** | H-bridge outputs |
| **22** | Universal I/O: IN 20/PROTECTED OUT 15 | **46** | Ground return: IN 20/PROTECTED OUT 15 | **70** | H-bridge outputs |

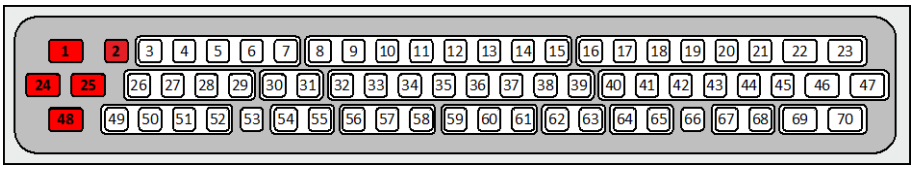
| Pin | Description | Pin | Description | Pin | Description |
|---|---|---|---|---|---|
| **23** | Universal I/O:<br>IN 21/PROTECTED OUT 16 | **47** | Ground return:<br>IN 21/PROTECTED OUT 16 | | |
| **24** | STANDARD FEED<br>(+12 VDC or +24 VDC) | **48** | Weight | | |

# Connecting the power supply and the 5 V output

**Introduction**

The following diagrams show the pinout of the connector (as viewed from the front):

**Power supply**




| Pin | Description | Terminal number in vehicles |
|---|---|---|
| 1 | PROTECTED FEED<br>(+12 VDC or +24 VDC) | Terminal # 30 |
| 2 | Ignition (+) (IGNITION FEED) | Terminal # 15 |
| 24 | STANDARD FEED<br>(+12 VDC or +24 VDC) | Terminal # 30 |
| 25 | Weight | Terminal # 31 |
| 48 | Weight | Terminal # 31 |

**Ignition (IGNITION FEED)**

- IGNITION FEED sources the digital electronics that control the inputs and outputs.
- Ignition must be on for the JXM-IO-E02 to be active.
- The JXM-IO-E02 will continue to run on a minimum input voltage of 5.9 V (on IGNITION FEED) in order to survive engine cranking (ISO 7637-2 Test Pulse 5 compliant).
  In general, the following conditions apply: The JXM-IO-E02 is designed to work with an input power voltage range of 8 V up to 32 V.
- The maximum current draw on this line is 2 A.

**STANDARD FEED**

- STANDARD FEED provides power for some of the outputs of the JXM-IO-E02.
- The maximum current draw on this line is 20 A.
- The input current on STANDARD FEED is monitored by the JXM-IO-E02.

**PROTECTED FEED**

- PROTECTED FEED provides power for some of the outputs of the JXM-IO-E02.
- The maximum current draw on this line is 20 A.
- If an error occurs, the internal protection circuits switch off the outputs. PROTECTED FEED is protected by solid-state switches.
  If the current current consumption exceeds 30 A or if one of the microcontrollers is defective, the solid-state switches are disabled by the redundant microcontrollers.

**Note on ignition**

To launch the JXM-IO-E02, pin 2 (IGNITION FEED) must be connected with pin 24 (STANDARD FEED). The ignition control signal is issued when the key is in position "Ignition ON".

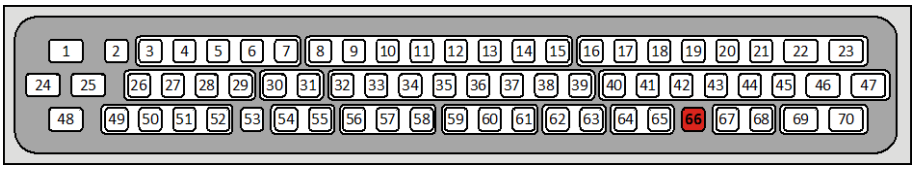
**Power Supply - Technical Data**

| Parameter | Description |
|---|---|
| Rated voltage | DC 12 V or DC 24 V |
| Permissible voltage range | DC 8 ... 32 V |
| Current consumption at 12 V (without load) | 120 mA |
| Current consumption at 24 V (without load) | 100 mA |

**Note on current consumption**

The base current consumption is measured shortly after switching on the JXM-IO-E02 while there are no active output signals and input signals are not connected. Active outputs and also certain connected input signals will affect the current consumption.

**Regulated 5 V output**




| Pin | Description |
|---|---|
| **66** | Regulated 5 V output |

**Technical data - Regulated output**

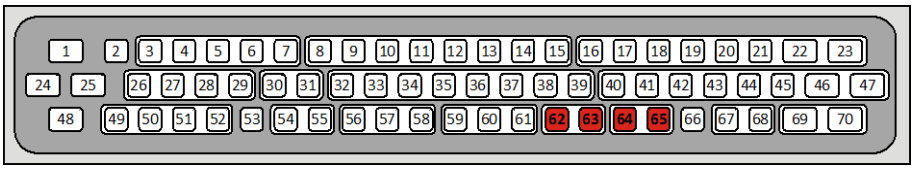
| Parameter | Description |
|---|---|
| Controlled voltage | DC 5 V |
| Load current | 250 mA max. |
| Over-current detection | Yes |

# CAN Interface and Node ID

**Introduction**

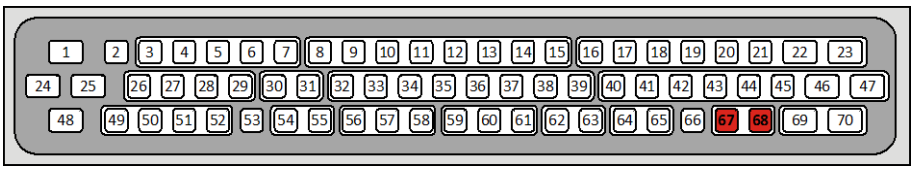
The following diagrams show the pinout of the connector (as viewed from the front):

**CANopen®**




| Pin | Description |
|---|---|
| **62** | OUT_CAN_L |
| **63** | OUT_CAN_H |
| **64** | IN_CAN_L |
| **65** | IN_CAN_H |

**Node ID**




| Pin | Description |
|---|---|
| **67** | Node ID (tri-state input # 1) |
| **68** | Node ID (tri-state input # 2) |

**Technical data - Tri-state inputs**

| Parameter | Description |
|---|---|
| Purpose | ■ For device coding<br>■ As digital inputs |
| Type of inputs | Pull-up resistor 22 kΩto IGNITION FEED and pull-down resistor 276 Ω to ground |
| Tri-state detection | Tri-state operation is detected by a pull-down resistor to ground. |
| Rated voltage | IGNITION FEED |
| Operating point OFF: | ≤ 1.0 V |
| Operating point ON: | ≥ 4.0 V |

**Note**

Note that because these inputs are tri-state enabled, they will always have bias voltage on the pin capable of sourcing current.

**Calculating the node ID based on tri-state input state**

The following table shows the resulting node ID given that the default base ID of 0x10 is used:

| State of pin 67 | State of pin 68 | CANopen® Node ID |
|---|---|---|
| Not connected | Not connected | 0x10 |
| Not connected | OFF | 0x11 |
| Not connected | ON | 0x12 |
| OFF | Not connected | 0x13 |
| OFF | OFF | 0x14 |
| OFF | ON | 0x15 |
| ON | Not connected | 0x16 |
| ON | OFF | 0x17 |
| ON | ON | 0x18 |

**Related topics**

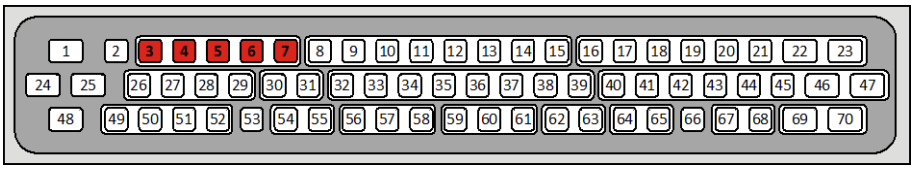
- **Specification - CAN bus cable** (see page 29)

# Connecting Digital Inputs and Outputs

**Introduction**

The following diagrams show the pinout of the connector (as viewed from the front):

**Digital inputs**

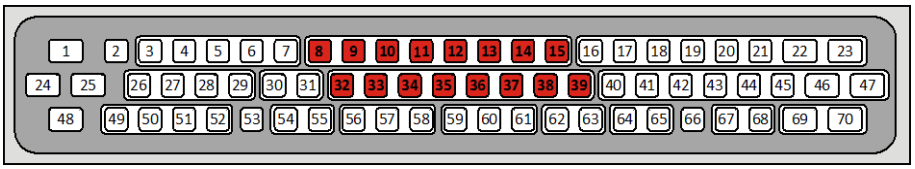



| Pin | Description |
|---|---|
| **3** | Digital input 1 |
| **4** | Digital input 2 |
| **5** | Digital input 3 |
| **6** | Digital input 4 |
| **7** | Digital input 5 |

**Technical data - Digital inputs IN 1 ... IN 5**

| Parameter | Description |
|---|---|
| Type of inputs | Software selectable with either 2 kΩ pull-up resistor to IGNITION FEED or 2 kΩ pull-down resistor to ground. |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Operating point OFF: | ≤ 1.0 V |
| Operating point ON: | ≥ 3.5 V |

**Digital Universal I/Os (STANDARD)**




| Pin | Description |
|---|---|
| **8** | Universal I/O: IN 6/OUT 1 |
| **9** | Universal I/O: IN 7/OUT 2 |
| **10** | Universal I/O: IN 8/OUT 3 |
| **11** | Universal I/O: IN 9/OUT 4 |
| **12** | Universal I/O: IN 10/OUT 5 |
| **13** | Universal I/O: IN 11/OUT 6 |
| **14** | Universal I/O: IN 12/OUT 7 |
| **15** | Universal I/O: IN 13/OUT 8 |

| Pin | Description |
|---|---|
| **32** | Ground return: IN 6/OUT 1 |
| **33** | Ground return: IN 7/OUT 2 |
| **34** | Ground return: IN 8/OUT 3 |
| **35** | Ground return: IN 9/OUT 4 |
| **36** | Ground return: IN 10/OUT 5 |
| **37** | Ground return: IN 11/OUT 6 |
| **38** | Ground return: IN 12/OUT 7 |
| **39** | Ground return: IN 13/OUT 8 |

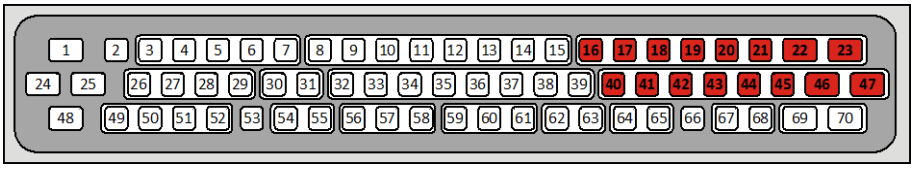
**Technical data - Digital inputs IN 6 ... IN 13**

| Parameter | Description |
|---|---|
| Type of inputs | Can be configured as active-high inputs |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Operating point OFF: | < 51 % of IGNITION FEED |
| Operating point ON: | > 51 % of IGNITION FEED |
| Input impedance | 100 kΩ |

**Technical data - Digital outputs (STANDARD FEED)**

| Parameter | Description |
|---|---|
| Type of outputs | Active-high output |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Signal voltage OFF | < 1.0 V |
| Signal voltage ON | $U_{STANDARD}$ - 0.5 V |
| Load current of OUT 1 ... OUT 8 | 2.5 A max. |
| Short-circuit capability | Yes |
| Over-current detection | Yes |
| No-load detection | Yes |

**Digital universal I/Os (PROTECTED)**




| Pin | Description |
|---|---|
| **16** | Universal I/O: IN 14/PROTECTED OUT 9 |
| **17** | Universal I/O: IN 15/PROTECTED OUT 10 |
| **18** | Universal I/O: IN 16/PROTECTED OUT 11 |
| **19** | Universal I/O: IN 17/PROTECTED OUT 12 |
| **20** | Universal I/O: IN 18/PROTECTED OUT 13 |
| **21** | Universal I/O: IN 19/PROTECTED OUT 14 |
| **22** | Universal I/O: IN 20/PROTECTED OUT 15 |
| **23** | Universal I/O: IN 21/PROTECTED OUT 16 |
| **40** | Ground return: IN 14/PROTECTED OUT 9 |
| **41** | Ground return: IN 15/PROTECTED OUT 10 |
| **42** | Ground return: IN 16/PROTECTED OUT 11 |
| **43** | Ground return: IN 17/PROTECTED OUT 12 |
| **44** | Ground return: IN 18/PROTECTED OUT 13 |
| **45** | Ground return: IN 19/PROTECTED OUT 14 |
| **46** | Ground return: IN 20/PROTECTED OUT 15 |
| **47** | Ground return: IN 21/PROTECTED OUT 16 |

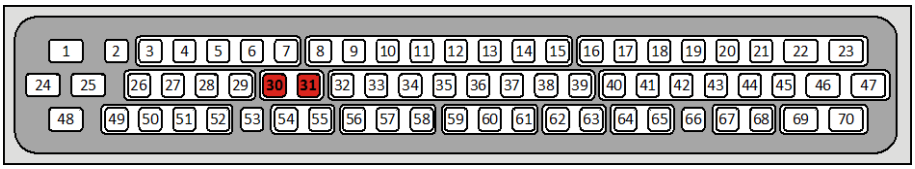
**Technical data - Digital inputs IN 14 ... IN 21**

| Parameter | Description |
|---|---|
| Type of inputs | Can be configured as active-high inputs |
| Rated voltage | PROTECTED FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Operating point OFF: | < 51 % of IGNITION FEED |
| Operating point ON: | > 51 % of IGNITION FEED |
| Input impedance | 100 kΩ |

**Technical data - Digital outputs (PROTECTED)**

| Parameter | Description |
|---|---|
| Type of outputs | Active-high output |
| Rated voltage | PROTECTED FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Signal voltage OFF | < 1.0 V |
| Signal voltage ON | $U_{PROTECTED}$ - 0.5 V |
| Load current of PROTECTED OUT 9 ... PROTECTED OUT 16 | 5.0 A max. |
| Can be switched off by electronic safety switch | Yes |
| Short-circuit capability | Yes |
| Over-current detection | Yes |
| No-load detection | Yes |

**Switch feed outputs**

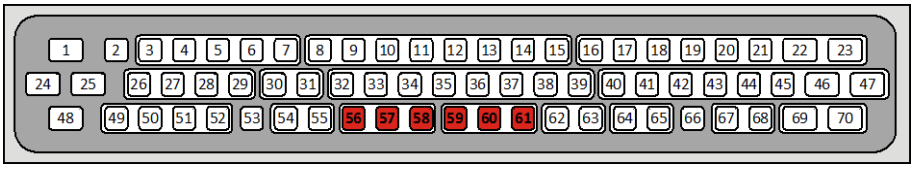



| Pin | Description |
|---|---|
| **30** | Switch feed output 1 |
| **31** | Switch feed output 2 |

**Technical data - Switch outputs**

| Parameter | Description |
|---|---|
| Type of switch outputs | Active-high output |
| Possible use | As digital active-low inputs |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Signal voltage OFF | < 1.0 V |
| Signal voltage ON | $U_{STANDARD}$ - 0.5 V |
| Load current | Each 2.5 A max. |
| Short-circuit capability | Yes, thermal (undefined current) |
| Over-current detection | Detection whether the output has switched off. |
| No-load detection | Yes |

## PWM outputs




| Pin | Description |
|---|---|
| **56** | PWM output 1 |
| **57** | PWM output 2 |
| **58** | PWM output 3 |
| **59** | Ground return: PWM output 1 |
| **60** | Ground return: PWM output 2 |
| **61** | Ground return: PWM output 3 |

## Technical data - PWM outputs

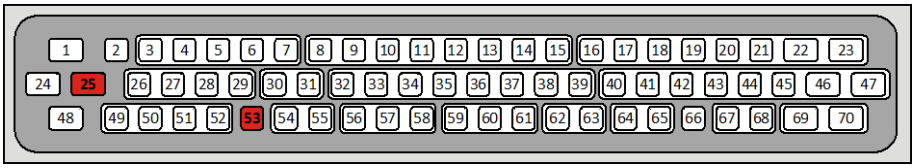
| Parameter | Description |
|---|---|
| Operating Modes | ■ Current-controlled output<br>■ PWM output with static duty cycle |
| PWM frequency | Configurable values:<br>0 = 1.954 kHz<br>1 = 977 Hz<br>2 = 488 Hz<br>3 = 244 Hz<br>4 = 122 Hz<br>5 = 61 Hz<br>6 = 30 Hz |
| Resolution | 10 bits |
| Load current | 0 ... 2.5 A |
| Short-circuit capability | Yes |
| Over-current detection | Yes |
| No-load detection | Yes |

# Connecting Analog Inputs and Outputs

**Introduction**

The following diagrams show the pinout of the connector (as viewed from the front):

**Analog output**

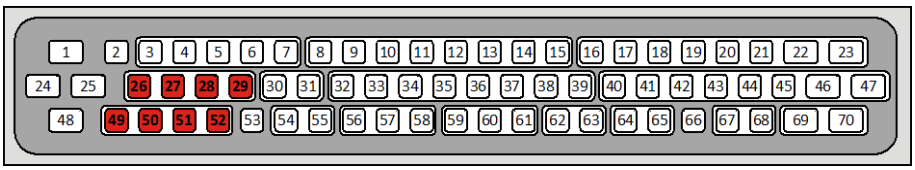



| Pin | Description |
|---|---|
| **25** | Weight |
| **53** | Analog output |

**Technical data - Analog output**

| Parameter | Description |
|---|---|
| Voltage range at 50 mA | 0 ... STANDARD FEED |
| Current range | 0 ... 100 mA |
| Resolution | 10 bits |
| Electrical isolation | None |
| Short circuit detection | Yes |

**Analog inputs**

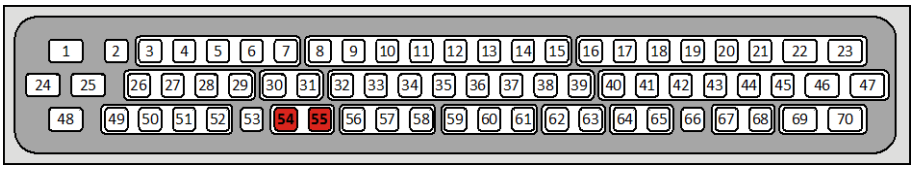



| Pin | Description |
|---|---|
| **26** | Ground: Analog input 1 |
| **27** | Ground: Analog input 2 |
| **28** | Ground: Analog input 3 |
| **29** | Ground: Analog input 4 |
| **49** | Analog input # 1 |
| **50** | Analog input # 2 |
| **51** | Analog input # 3 |
| **52** | Analog input # 4 |

### Technical data - Analog inputs

| Parameter | Description |
|---|---|
| Voltage range | ■ 0 ... 5 V<br>■ 0 ... 40 V |
| Current range | ■ 0 ... 20 mA |
| Input impedance at 0 ... 5 V | 100 kΩ |
| Input impedance at 0 ... IGNITION FEED | 50 kΩ |
| Input impedance at 0 ... 20 mA | 240 Ω |
| Resolution | 10 bits |
| Electrical isolation | None |

### Frequency inputs

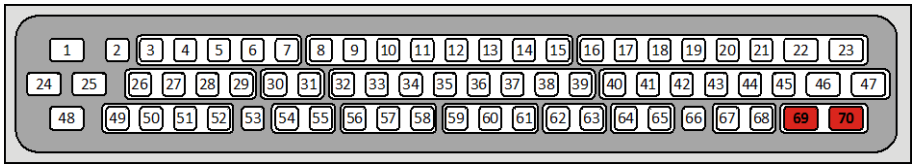



| Pin | Description |
|---|---|
| **54** | Frequency input # 1 |
| **55** | Frequency input # 2 |

### Technical Data - Frequency Inputs

| Parameter | Description |
|---|---|
| Purpose | ■ As frequency counter<br>■ As digital input |
| Type of inputs | Software selectable with either 2 kΩ pull-up resistor to STANDARD FEED or 2 kΩ pull-down resistor to ground. |
| Frequency measurement range | 5 Hz ... 20 kHz |
| Measurement method | Time-based |
| Result of measurement | Period of the signal in nanoseconds |
| Resolution | 62.5 ns |
| As of hardware revision ≥ 12.04 the following operating point parameters apply: | |
| Operating point OFF: | < 1.7 V for mode 1 ... 4 and IGNITION FEED = 12 V |
| Operating point ON: | > 3.6 V for mode 1 ... 4 and IGNITION FEED = 12 V |
| Operating point OFF: | < 3.4 V for mode 1 ... 4 and IGNITION FEED = 24 V |
| Operating point ON: | > 7.2 V for mode 1 ... 4 and IGNITION FEED = 24 V |
| Operating point OFF: | AC signal < -350 mV for mode 5 |
| Operating point ON: | AC signal > 350 mV for mode 5 |

### H-bridge outputs




| Pin | Description |
|---|---|
| **69** | H-bridge outputs |
| **70** | |

**Technical data - H-bridge**

| Parameter | Description |
|---|---|
| Purpose | ■ As H-bridge<br>■ As two independent digital inputs<br>■ As two independent PWM outputs (active-high and active-low) |
| Rated output current | 2.5 A max. |
| Accuracy of current measurement for H-bridge | < 100 mA |
| Short-circuit capability | Yes |
| Overcurrent detection | Yes |
| No-load detection | Yes |

# 4.2 Installing the JXM-IO-E02

**Introduction**

This chapter describes how to install the JXM-IO-E02.

**Contents**


| Topic | Page |
|---|---|
| Installing the JXM-IO-E02 | 46 |

## Installing the JXM-IO-E02

**Selecting a place for installation**

Select a suitable place for the device to be installed.
A place is suitable if it fulfills the following requirements:

- The installation surface must be made from one of the following materials:
  - Aluminum plate
  - Galvanized steel plate
  - Painted steel plate
- The installation surface must be vertical.
- The installation surface must be level.
- The installation location must allow adequate air circulation.
- The installation location must be accessible for servicing.
- The installation location must be of sufficient size.

For further information refer to **Physical Dimensions** (see page 17)

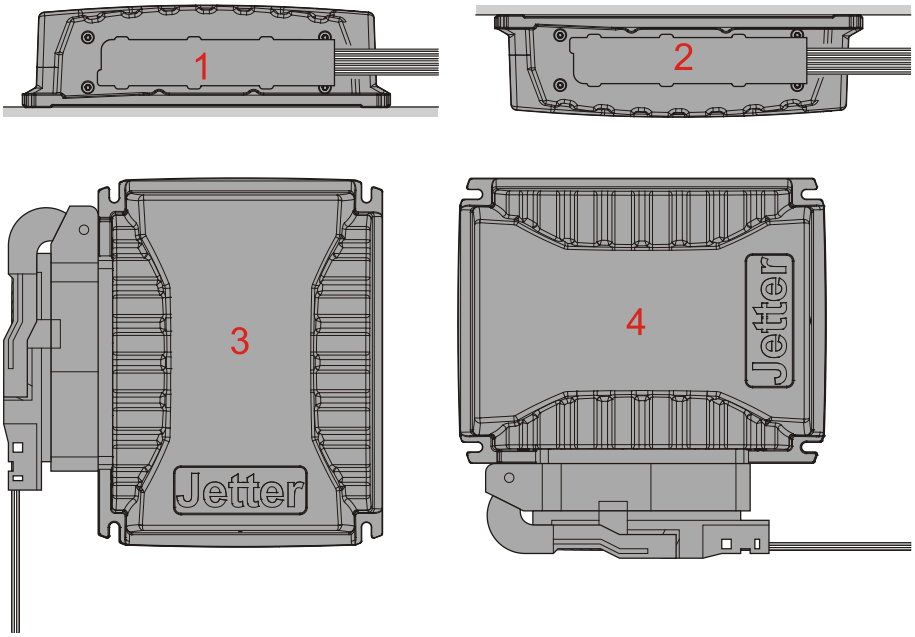
**Avoiding unsuitable installation locations**

Do not install the device in locations that do not meet the a.m. requirements.
The following locations are not appropriate for installing the JXM-IO-E02:

| Inappropriate location | Reason |
|---|---|
| Unventilated installation location | The device could overheat as heat builds up. |
| Stainless steel surfaces | Galvanic corrosion may occur between device and mounting surface |
| Installation location close to heat-sensitive materials | The materials could become warped or misshapen as a result of heat produced by the device. |
| Uneven installation surfaces | The installation surface could become misshapen when fitting the device.<br>Fastening is unstable and precarious. |

**Permitted installation positions**

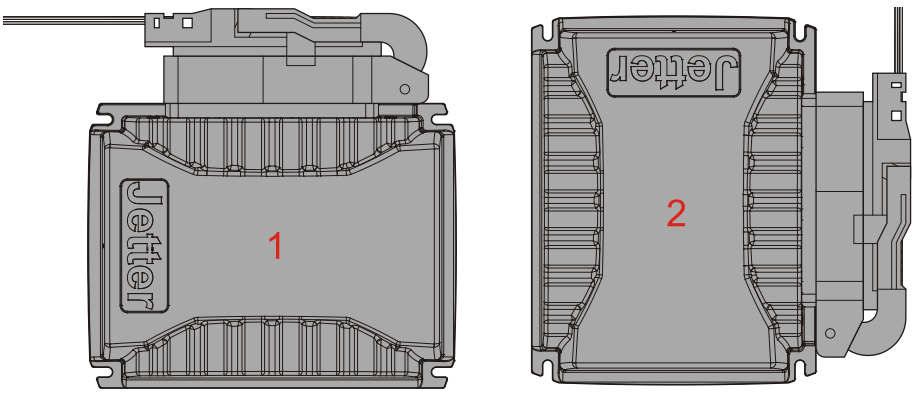
The illustration shows the positions permitted for installation.




| Number | Permitted installation position |
|---|---|
| **1** | Horizontally, lying |
| **2** | Horizontally, hanging |
| **3** | Vertically, connector left |
| **4** | Vertically, connector downwards |

**Prohibited installation positions**

The diagram shows the positions prohibited for installation.




| Number | Prohibited Installation Positions |
|---|---|
| **1** | Vertically, connector upwards |
| **2** | Vertically, pressure equalizing membrane upwards |

Why are these installation positions prohibited?

- Vertically, connector upwards: The accumulation of moisture and water droplets in the connector can lead to current leakages and corrosion.
- Vertically, pressure equalizing membrane upwards: The accumulation of moisture and water droplets can block the hole which may impede pressure compensation.

**Selecting installation hardware**

For installation use the following hardware:

| Part | Description |
|---|---|
| Screws/bolts | Size: M 5 x 15<br>Surface: Galvanized<br>Strength class: 8.8 |
| Washers | Size: 5.3 x 10<br>Surface: Galvanized |
| Screw nuts | Size: M 5<br>Surface: Galvanized<br>Strength class: 8.8 |

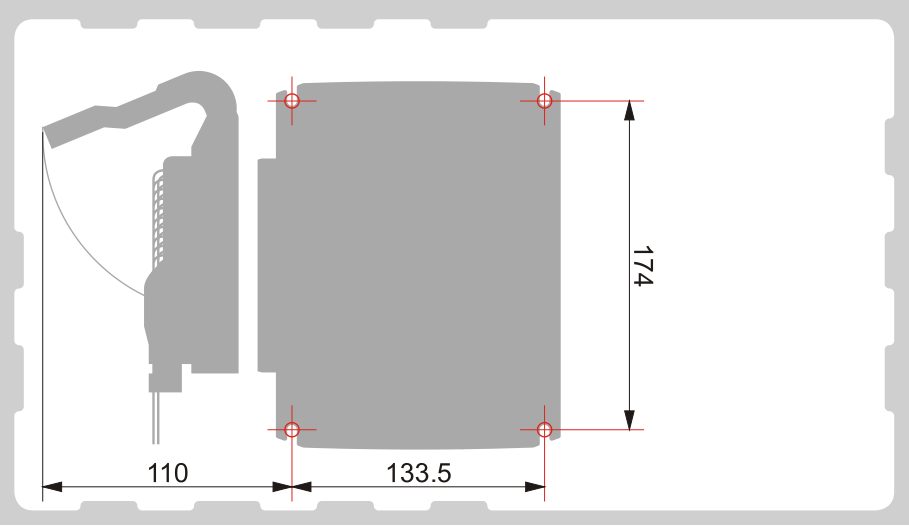
**Avoid Improper Installation Material**

You must avoid installation material made from stainless steel. In connection with the housing material of the JXM-IO-E02 galvanic corrosion may occur.

**Preparing for installation**

Mark off the positions of the four mounting holes.

Center-punch the four holes.




| If ... | ... then ... |
|---|---|
| ... the thickness of the mounting surface is ≥ 6 mm (steel) and ≥ 8 mm (aluminum), ... | ... drill tapped holes:<br>■ Pre-drill Ø 4.2 mm.<br>■ Tap a thread M 5. |

| If ... | ... then ... |
|---|---|
| ... the thickness of the mounting surface is < 6 mm (steel) and < 8 mm (aluminum), ... | ... drill simple holes:<br>■ Drill the holes Ø 6 mm.<br>■ Deburr the holes. |

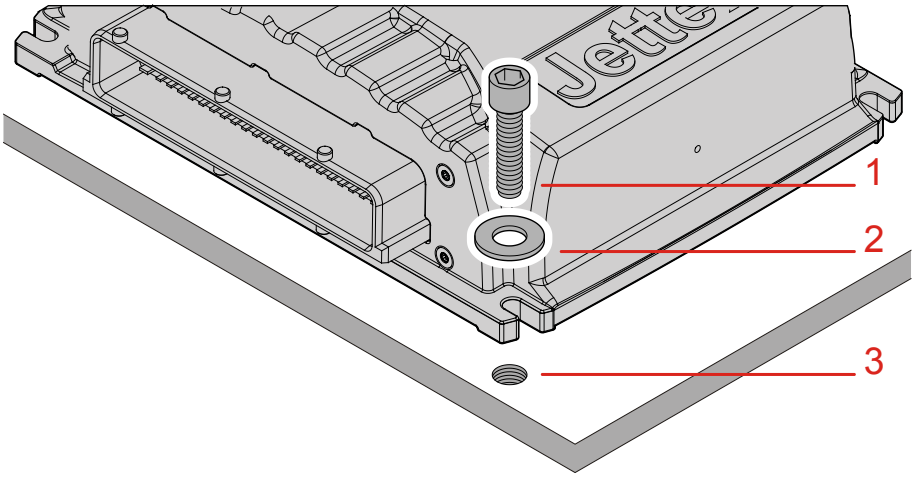
**Notes on installation**

Direct contact between housing and installation surface improves heat dissipation. Therefore:

- Generally install the device directly on the installation surface.
- Do not use insulating material.
- Do not use spacers.

**Installing the JXM-IO-E02 (tapped holes)**

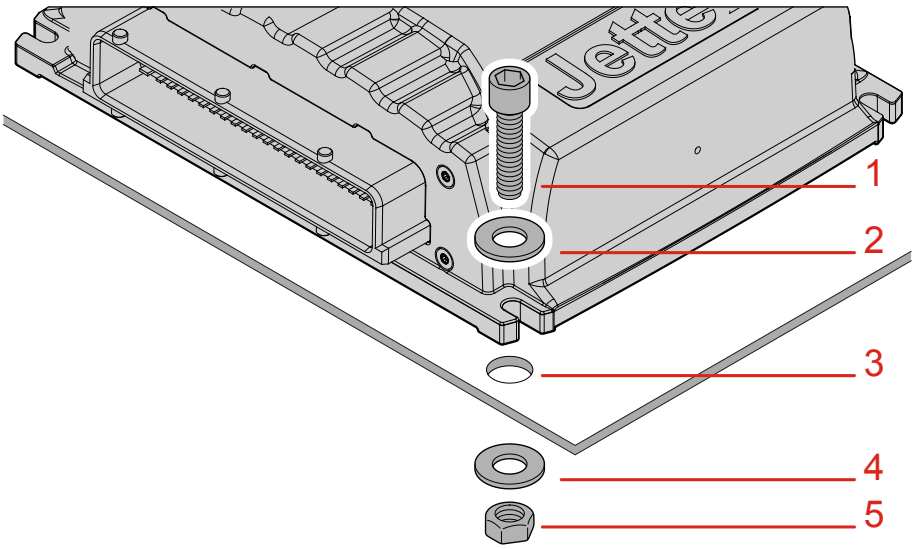
Screw the device down to the installation surface.




| Number | Description |
|---|---|
| **1** | Screw |
| **2** | Washer |
| **3** | Tapped hole |

**Installing the JXM-IO-E02 (through holes)**

Screw the device down to the installation surface.



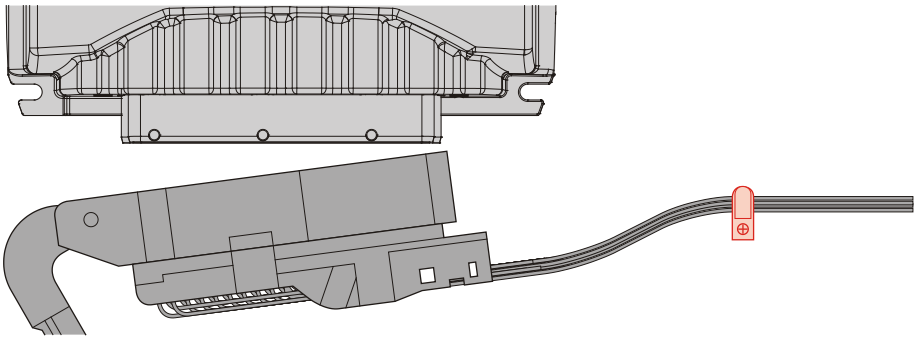

| Number | Description |
|---|---|
| **1** | Screw |
| **2** | Washer |
| **3** | Through hole |
| **4** | Washer |
| **5** | Screw nut |

**Installing the strain relief**

Install a strain relief for the connection cable.

Take care to leave enough space for the connector.

The connector should not be obstructed, so that it can be removed in the event of a service requirement.



# 5 Initial commissioning

**Introduction**

This chapter describes how to commission the JXM-IO-E02 and covers the following steps:

- Connecting the power supply and interfaces
- Initial commissioning via CANopen® interface

**Contents**


| Topic | Page |
|---|---|
| Preparatory work for initial commissioning | 52 |
| Information on communication with a JXM-IO-E02 | 54 |

# Preparatory work for initial commissioning

**Introduction**

To be able to commission and program the JXM-IO-E02, complete the following activities first:

- Connecting a controller with power supply, ignition and CAN interface.
- Changing the node ID in the case of several CANopen® nodes of the same module type.

**Default values**

Some of the default values of the JXM-IO-E02 module are listed below:

- Baud rate: 250 kBaud
- CAN terminating resistor: 0x01
  This value means that the CAN terminating resistor of 120 Ω at the end of the CAN bus is enabled.
- Node ID: 0x10

**Wiring**

To wire the module JXM-IO-E02, proceed as follows:

| Step | Action |
|---|---|
| **1** | Connect the following terminals with the power supply DC 8 - 32 V:<br>■ PROTECTED FEED **Pin 1** (terminal 30 in the vehicle)<br>■ IGNITION **Pin 2** (terminal 15 in the vehicle)<br>■ STANDARD FEED **Pin 24** (terminal 30 in the vehicle)<br>■ GROUND **Pin 25** (terminal 31 in the vehicle) |
| **2** | Connect the module to the CANopen® bus (**pin 62** and **pin 63, pin 64** and **pin 65**). |
| **3** | Make sure that there is a terminating resistor of 120 Ω at both ends of the CAN bus. |
| **4** | Switch on the ignition to energize the module.<br>Communication with the module is now possible. |

**Result:** The module is now operational and can be initialized by the controller.

**Data collision in the case of several modules with the same node ID.**

If you connect several nodes of the same type to the CANopen® bus, this leads to data collisions on the CANopen® bus. The reason for this is that modules of the same type in as delivered condition have got the same node ID. To allow communication with all nodes on the bus, the node ID of modules with the same default ID must be changed. There are two ways to change the node ID:

**Option 1:**

Connect modules with different states of the tri-state inputs. The resulting node IDs are as follows:

| Module | State of pin 67 | State of pin 68 | CANopen® Node ID |
|---|---|---|---|
| Module 1 | Not connected | Not connected | 0x10 |
| Module 2 | Not connected | OFF<br>(terminal 31 in the vehicle) | 0x11 |
| ... | ... | ... | ... |
| Module 9 | ON<br>(terminal 15 in the vehicle) | Not connected | 0x16 |

**Option 2:**

Connect the modules one after the other to the CANopen® bus. Then, change the node ID through the system parameters as described below:

| Step | Action | Result |
|---|---|---|
| **1** | Connect module 1 to the CANopen® bus. | |
| **2** | Enter a new node ID into the system parameter with index 0x4556 (sub-index 4). | As a result, module 1 has got a new node ID. |
| **3** | Then, connect module 2 to the CANopen® bus. | |
| **4** | Enter a new node ID into the system parameter with index 0x4556 (sub-index 4). | As a result, module 2 has got a new node ID. |
| **5** | Repeat this procedure for all modules of the same type until all modules are connected to the CANopen® bus. | Now, collision of data is avoided in communication between nodes of the same type connected to the CANopen® bus since each node has got an individual node ID. |

**Related topics:**

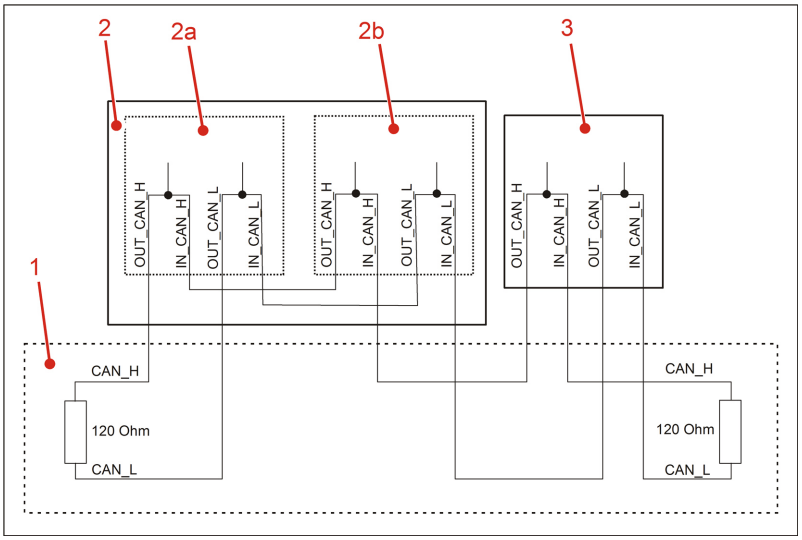
- **Wiring** (see page 26)
- **Information on communication with the module** (see page 54)
- **CANopen® interface and node ID** (see page 35)

# Information on communication with a JXM-IO-E02

**Wiring diagram - Example**

The illustration below shows a wiring example of the following CANopen® devices by Jetter AG:

- Controller JCM-350-E03
- Peripheral module JXM-IO-E02




| Number | Description | Node ID |
|---|---|---|
| **1** | CAN bus | |
| **2** | Jetter controller JCM-350-E03 | |
| **2a** | Controller JCM-350 | 0x7F (127 decimal) |
| **2b** | I/O module JXM-IO-E02 | 0x10 (16 decimal) |
| **3** | Separate I/O module JXM-IO-E02 | 0x11 (17 decimal) for tri-state inputs that have been configured by the user |

**CANopen® interface - Restrictions**

During initial commissioning take into account the following restrictions/limitations of the CANopen® interface on the JXM-IO-E02:

- PDOs are not user configurable.
- PDOs are transmitted only asynchronously on request.

**Communication with peripheral modules**

The following information supports you in commissioning peripheral modules, such as JXM-IO-E02:

- Initialize the controller as described in the documentation on JCM-350-E03.
- Send an RTR frame to the peripheral module. This parameter is needed once in order to prompt the peripheral module to send the required data to the controller.

**JetSym STX sample program**

The following sample program shows how the states of the digital inputs on the JXM-IO-E02 can be read by a Jetter controller, such as JCM-350.

```
Const
    CAN_CONTROLLER_0 = 0;

    //Node ID of the controller
    NodeID_Node_0 = 0x7F;

    //Node ID of the I/O module
    NodeID_Node_1 = 0x10;

    Event_Time = 100;
    Inhibit_Time = 20;
End_Const;

Var
//State of the digital inputs
    Data_Inputs: Word;
    SW_Version: String;
End_Var;


Task Main Autorun

// Software version of the controller
SW_Version := 'v4.3.0';

// Initializing CAN 0

CanOpenInit(CAN_CONTROLLER_0, NodeID_Node_0, SW_Version);

// Entering process data to be received
CanOpenAddPDORx(CAN_CONTROLLER_0,
CANOPEN_PDO1_RX(NodeID_Node_1), 2, CANOPEN_WORD,
sizeof(Data_Inputs), Data_Inputs, Event_Time, Inhibit_Time,
CANOPEN_ASYNCPDORTRONLY);

// All nodes on the CAN bus are in PREOPERATIONAL state
// Setting all devices on the CAN bus to OPERATIONAL status
CanOpenSetCommand(CAN_CONTROLLER_0,
CAN_CMD_NMT_Value(CAN_CMD_NMT_ALLNODES, CAN_CMD_NMT),
CAN_NMT_START);

End_Task;
```

**Related topics:**

- **CANopen® objects** (see page 57)

# 6 CANopen® objects

**Introduction**

This chapter covers the CANopen® objects implemented on the JXM-IO-E02 and their functions, as well as the permanently mapped process data objects (PDO).

**Restrictions**

Due to design constraints the following restrictions/limitations apply to the CANopen® interface of the JXM-IO-E02:

- SDO expedited transfer only supports 4-byte transfers. Any smaller data element must be extended to 32 bit before the SDO transfer.
- SDO segmented transfer is only supported on certain objects. Most notably the OS update feature makes use of segmented transfer, but also some other objects that need to transfer strings implement SDO segmented transfer for this purpose. Unless an object is documented to support segmented transfers, assume that it does not.
- SDO block transfer is NOT implemented.
- PDOs are not user configurable.
- PDOs are transmitted only asynchronous on request unless otherwise specified.
- Although emergency messages are transmitted to notify of detected faults, the CANopen® emergency handling system is not fully implemented.
- The Error Register does not save its state in non volatile memory. After each reset or power cycle, the error list is cleared.

**Contents**


| Topic | Page |
|---|---|
| CANopen® object dictionary for JXM-IO-E02 | 58 |
| CANopen® PDO Specification | 108 |

# 6.1 CANopen® object dictionary for JXM-IO-E02

**Purpose of this chapter**

This chapter describes the CANopen® objects implemented on the JXM-IO-E02 and their function.

**Supported objects**

There are objects which are mandatory according to the CANopen® specification. These objects are not covered in this document. The table below lists the objects which are covered in this document.

| Index (hex) | Object name | Object (code) | Type | see |
|---|---|---|---|---|
| 1000 | Device type | VAR | Unsigned32 | **Page** 60 |
| 1001 | Error Register | VAR | Unsigned8 | **Page** 60 |
| 1003 | Pre-defined error field | ARRAY | Unsigned32 | **Page** 60 |
| 100A | Manufacturer software version | VAR | String | **Page** 60 |
| 1017 | Producer heartbeat time | VAR | Unsigned16 | **Page** 60 |
| 1018 | Identity | RECORD | Identity (23h) | - |
| 2000 | Features | ARRAY | Unsigned32 | **Page** 60 |
| 2100 | Digital Inputs | ARRAY | Unsigned32 | **Page** 62 |
| 2101 | Universal I/O | ARRAY | Unsigned32 | **Page** 64 |
| 2102 | Tri-state Inputs | ARRAY | Unsigned32 | **Page** 66 |
| 2103 | Switch Feed Outputs | ARRAY | Unsigned32 | **Page** 68 |
| 2200 - 2203 | Analog Input | ARRAY | Unsigned32 | **Page** 69 |
| 2210 | Voltage Sense Analog Input | ARRAY | Unsigned32 | **Page** 71 |
| 2211 | Feed Currents | ARRAY | Unsigned32 | **Page** 72 |
| 2300 | Analog Output | ARRAY | Unsigned32 | **Page** 73 |
| 2400 - 2402 | PWM Output | ARRAY | Unsigned32 | **Page** 75 |
| 2500 | H-Bridge | ARRAY | Unsigned32 | **Page** 80 |
| 2600, 2601 | Frequency Input | ARRAY | Unsigned32 | **Page** 82 |
| 2800 | 5 V Output | ARRAY | Unsigned32 | **Page** 84 |
| 4554 | OS Update | ARRAY | Unsigned32 | **Page** 85 |
| 4555 | Electronic Datasheet | ARRAY | Unsigned32 | **Page** 85 |
| 4556 | System Parameters | ARRAY | Unsigned32 | **Page** 86 |
| 4557 | OS Status | ARRAY | Unsigned32 | **Page** 101 |
| 4559 | Detailed Software Version | ARRAY | Unsigned32 | **Page** 102 |
| 4560 | Slave CAN termination | ARRAY | Unsigned32 | **Page** 103 |
| 4561 | Master CAN termination | ARRAY | Unsigned32 | **Page** 104 |
| 4565 | ENP SDO | ARRAY | Unsigned32 | **Page** 105 |
| 5000 | User EEPROM Access | ARRAY | Unsigned32 | **Page** 106 |

**Contents**


| Topic | Page |
|---|---|
| Objects ranging from index 0x1000 through 0x2000 | 60 |
| Object "Digital Outputs" (Index 0x2100) | 62 |
| Objekt " Digital Universal-I/O" (Index 0x2101) | 64 |
| Object "Tri-state Input" (Index 0x2102) | 66 |
| Object "Switch Feed Output" (Index 0x2103) | 68 |
| Objects "Analog Input" (Index 0x2200 ... 0x2203) | 69 |
| Object "Voltage Sense Analog Input" (Index 0x2210) | 71 |
| Object "Feed Currents" (Index 0x2211) | 72 |
| Object "Analog Output" (Index 0x2300) | 73 |
| Objects "PWM Output" (Index 0x2400 ... 0x2402) | 75 |
| Object "H-Bridge" (index 0x2500) | 80 |
| Objects "Frequency Input" (Index 0x2600 ... 0x2601) | 82 |
| Object "5 V Output" (0x2800) | 84 |
| Objekte "OS Update" (Index 0x4554) und EDS (Index 0x4555) | 85 |
| Object "System Parameter" (Index 0x4556) | 86 |
| Validating and setting system parameters | 98 |
| Object "OS Status" (Index 0x4557) | 101 |
| Object "Detailed Software Version" (Index 0x4559) | 102 |
| Object "Slave CAN Termination" (Index 0x4560) | 103 |
| Object "Master CAN Termination" (Index 0x4561) | 104 |
| Object "Electronic Name Plate" (Index 0x4565) | 105 |
| Object "User EEPROM Access" (Index 0x5000) | 106 |

## Objects ranging from index 0x1000 through 0x2000

**Device Type (Index 0x1000)**

The structure of the object "Device Type" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x1000 | 0 | 0x008F0191 | Type of device | ro (read only) |

This object is read-only and conforms with the CANopen® specification.

**Error Register (Index 0x1001)**

The structure of the object "Error Register" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x1001 | 0 | 0 | Error register | ro (read only) |

This object implements the CANopen® error register functionality.

Bit 0 = Generic error
Bit 1 = Current error
Bit 2 = Voltage error
Bit 3 = Temperature error
Bit 4 = Communication error
Bit 5 = Parameter mismatch
Bit 6 = Not applicable
Bit 7 = Manufacturer-specific error, for example, hardware error

**Pre-defined Error Field (Index 0x1003)**

The structure of the object "Pre-defined Error Field" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x1003 | 0 | 0 | Number of errors entered in the array's standard error field | rw (read & write) |
| | 1 | 0 | Most recent error; **0** indicates no error | ro (read only) |
| | 2 ... 64 | - | Earlier errors | ro |

This object shows a history list of errors that have been detected by the JXM-IO-E02. The maximum length of the list is 64 errors. The list content is deleted on restart.

As per CANopen® specification this list can be cleared by entering the value **0** into sub-index 0.

**Composition of Standard Error Field**

2-byte LSB: Error code

2-byte MSB: Additional information

**Manufacturer Software Version (Index 0x100A)**

The structure of the object "Manufacturer Software Version" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x100A | 0 | | Software version | const |

Use only the STX function `CanOpenUploadSDO()` to determine the version of the software running in the JXM-IO-E02.

The version string has a length of 9 characters and is of the format **2.00.0.00**. The first digit is the major revision followed by the minor revision and the branch and beta indicators (which will usually be zero). This value is read-only (ro).

**Producer Heartbeat Time (Index 0x1017)**

The structure of the object "Producer Heartbeat Time" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x1017 | 0 | 1,000 [ms] | Heartbeat time | rw (read & write) |

The legal range for values is 250 ... 65,535.

**Features (Index 0x2000)**

The structure of the object "Features" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2000 | 0 | | Features Object | ro (read only) |

This object is only provided for compatibility reasons.

## Object "Digital Outputs" (Index 0x2100)

**Digital Input (Index 0x2100)**

The structure of this object is shown in the table below. This object is for configuring the digital inputs IN 1 ... IN 5 and for retrieving their states.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2100 | 0 | 6 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Active-high/active-low selection | rw (read & write) |
| | 3 | 1 | Not used | |
| | 4 | 0 | Process value 0: Input states (2 bits/channel) | ro |
| | 5 | 0 | Process value 1: Input states (1 bit/channel) | ro |
| | 6 | 5 | Parameter 0: Amount of inputs | ro |

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-Index 2 lets you set inputs IN 1 through IN 5 to either active-high (internal pull down resistor) or active-low (internal pull up resistor) mode.
- A bit value of **0** sets the input to active-low (input state "OFF") and a bit value of **1** sets the input to active-high (input state "ON"). The value can also be read back to confirm.
- Sub-index 2 uses the one bit per channel data structure described below:
  - Bit 0: Digital input IN 1
  - Bit 1: Digital input IN 2
  - Bit 2: Digital input IN 3
  - Bit 3: Digital input IN 4
  - Bit 4: Digital input IN 5

**Sub-index 4**

The function of sub-index 4 is described below:

- Sub-index 4 lets you retrieve the current state of IN 1 through IN 5.
- Sub-index 4 returns the data in a two bit per channel format (provided for backwards compatibility).
- Sub-index 4 uses the two bit per channel data structure described below:
  - Bit 1, 0: Digital input IN 1
  - Bit 3, 2: Digital input IN 2
  - Bit 5, 4: Digital input IN 3
  - Bit 7, 6: Digital input IN 4
  - Bit 9, 8: Digital input IN 5
- In the two bit per channel configuration, the following data values are possible:
  - 0b00: Not used
  - 0b01: Input state OFF

| | |
|---|---|
| | • 0b10: Input state ON<br>• 0b11: Not used |
| **Sub-index 5** | The function of sub-index 5 is described below:<br>▪ Sub-index 5 lets you retrieve the current state of IN 1 through IN 5.<br>▪ Sub-index 5 returns the data in the one bit per channel data structure described below:<br>▪ Sub-index 5 uses the one bit per channel data structure described below:<br>• Bit 0: Digital input IN 1<br>• Bit 1: Digital input IN 2<br>• Bit 2: Digital input IN 3<br>• Bit 3: Digital input IN 4<br>• Bit 4: Digital input IN 5<br>▪ In the one bit per channel configuration, the following data values are possible:<br>• 0: Input state OFF<br>• 1: Input state ON |
| **Sub-index 6** | Sub-index 6 lets you retrieve the number of available inputs. In the given case, five inputs are available. |

## Objekt " Digital Universal-I/O" (Index 0x2101)

**Universal I/O**

A universal I/O can be used as digital input or digital output. Therefore, universal I/Os must be configured correspondingly.

- You can configure as many multi-purpose I/Os as required as digital input or output.
- If a universal I/O is used as digital input, the related digital output must be disabled (OFF).

**Universal I/O (Index 0x2101)**

This object lets you configure universal I/Os. It allows either to read out the state of the digital inputs **IN 6 ... IN 21**, or to set the digital outputs **OUT 1 ... 16**.

The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2101 | 0 | 6 | Number of entries | ro (read only) |
| | 1 | 0 | Enabling channel | rw (read & write) |
| | 2 | 0 | Disabling channel | rw |
| | 3 | 4 | Not used | |
| | 4 | 0 | Process value 0: Reading back output states / reading out input states | rw |
| | 5 | 0 | Process value 1: Output states | rw |
| | 6 | 16 | Parameter 0: Number of inputs/outputs | ro |

**Sub-index 1**

The function of sub-index 1 is described below:

- Sub-index 1 lets you enable individual channels.
- To enable a channel, enter its number (**1 ... 16**) into sub-index 1.
- Reading out sub-index 1 will always return the value **0**.

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 lets you disable individual channels.
- To disable a channel, enter its number (**1 ... 16**) into sub-index 2.
- Reading out sub-index 2 will always return the value **0**.

**Sub-index 4**

The function of sub-index 4 is described below:

- Sub-index 4 lets you retrieve the current state of **IN 6 ... IN 21**.
- In addition, it lets you read back the states of **OUT 1 ... OUT 15**.
- In sub-index 4, each bit is assigned to a channel:
  - Bit 0: Channel 1 (IN 6 or OUT 1)
  - Bit 1: Channel 2 (IN 7 or OUT 2)
  - ...
  - Bit 14: Channel 15 (IN 20 or OUT 15)
  - Bit 15: Channel 16 (IN 21 or OUT 16)
- If a universal I/O is used as digital input, the related digital output must be disabled (OFF).

**Sub-index 5**

The function of sub-index 5 is described below:

- Sub-index 5 lets you set or reset the digital outputs **OUT 1 ... OUT 16**.
- In sub-index 5, each bit is assigned to a channel:
  - Bit 0: Channel 1 (OUT 1)
  - Bit 1: Channel 2 (OUT 2)
  - ...
  - Bit 14: Channel 15 (OUT 15)
  - Bit 15: Channel 16 (OUT 16)
- Depending on the bit value, the output state is as follows:
  - 0: Output state OFF
  - 1: Output state ON

**Sub-index 6**

Sub-index 6 lets you read out the number of available inputs/outputs. In the given case, 16 inputs/outputs are available.

## Object "Tri-state Input" (Index 0x2102)

**Purpose of tri-state inputs**

Tri-state inputs are generally used for obtaining the node ID or changing the default node ID (device coding). However, in applications where device coding is not required, these inputs can be freely used as general-purpose digital inputs. This may be the case if only one JCM-350-E03 or JXM-IO-E02 is connected to the CAN bus.

Object **System Parameters** (Index 0x4556, sub-index 38) lets you disable the flag "Tri-state Coding Enable". To do so, enter the value **0**.

**Tri-state input (Index 0x2102)**

The object **Tri-State Inputs** is read-only. Read this object to obtain the states of the tri-state inputs. The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2102 | 0 | 6 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Not used | |
| | 3 | 3 | Not used | |
| | 4 | 0 | Process value 0: Input states | ro |
| | 5 | 0 | Process value 1: Not used | |
| | 6 | 2 | Parameter 0: Amount of inputs | ro |

**Sub-index 4**

Sub-index 4 lets you read out the latest measured states of the two tri-state inputs.

Because each input can be in one of three different input states, sub-index 4 uses the two bit per channel data structure described below:

- Bit 1, 0: Tri-state input 1 (pin 67)
- Bit 3, 2: Tri-state input 2 (pin 68)

The following values are allowed:

- 0b00: Tri-state (not connected)
- 0b01: Input state OFF
- 0b10: Input state ON
- 0b11: Not used

**Sub-index 6**

Sub-index 6 lets you read out the number of available inputs. In the given case, two inputs are available.

**Calculating the node ID based on tri-state input state**

The following table shows the resulting node ID given that the default base ID of 0x10 is used:

| State of pin 67 | State of pin 68 | CANopen® Node ID |
|---|---|---|
| Not connected | Not connected | 0x10 |
| Not connected | OFF | 0x11 |
| Not connected | ON | 0x12 |
| OFF | Not connected | 0x13 |
| OFF | OFF | 0x14 |
| OFF | ON | 0x15 |
| ON | Not connected | 0x16 |
| ON | OFF | 0x17 |
| ON | ON | 0x18 |

**Changing the default node ID 0x10**

The node ID is stored in the internal EEPROM and is read out during the boot process. For special applications, the object **System Parameters** (index 0x4556, sub-index 4) lets you change the value stored in the EEPROM.

## Object "Switch Feed Output" (Index 0x2103)

**Switch Feed Output (Index 0x2103)**

This object lets you enable or disable the two switch feed outputs. The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2103 | 0 | 6 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Not used | |
| | 3 | 4 | Not used | |
| | 4 | 0 | Process value 0: Information on the state | |
| | 5 | 0 | Process value 1: Output state | rw (read & write) |
| | 6 | 2 | Parameter 0: Number of outputs | ro |

**Sub-index 4**

Sub-index 4 returns the following status information on the switch feed outputs:

- Bit 0 represents switch feed output 1
- Bit 1 represents switch feed output 2
- The following bit values are possible:
  - 0: An error has occurred
  - 1: Normal operation, no error

**Sub-index 5**

The function of sub-index 5 is described below:

- Sub-Index 5 lets you enable/disable the output state of both switch feed channels.
- Sub-index 5 uses the one bit per channel data structure described below:
  - Bit 0 = Switch feed output 1
  - Bit 1 = Switch feed output 2
- The following values are allowed:
  - 0: Disable the switch feed output
  - 1: Enable the active-high switch feed output

**Sub-index 6**

Sub-index 6 lets you read out the number of available switch feed outputs.

**Digital active-low-side outputs**

Switch feed outputs can also be used as digital active-low-side outputs. If an output is disabled and no load is connected to it, then status bit in sub-index 4 is "0". If there is a low-resistance connection to ground, then status bit 4 is "1".

# Objects "Analog Input" (Index 0x2200 ... 0x2203)

**Analog Input (Index 0x2200 ... 0x2203)**

This object lets you configure the analog inputs 1 ... 4. The analog input signal can be read out as process value. The structure of these objects is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2200 ... 0x2203 | 0 | 7 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Operating mode | rw (read & write) |
| | 3 | 0x30 | Not used | |
| | 4 | 0 | Process value 0: Analog input signal | ro |
| | 5 | 0 | Process value 1: Analog input signal [mV] | ro |
| | 6 | 8,191 | Parameter 0: Maximum output value | ro |
| | 7 | 40,000 | Parameter 1: Maximum output value | ro |

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 lets you select between two modes of operation. One of these modes supports Automatic Gain Control (AGC). The other mode of operation supports either voltage measurement or current measurement.
- To select the operating mode, set or reset bit 0 and bit 4:
  - Bit 0 = 0: Disable AGC
  - Bit 0 = 1: Enable AGC
  - Bit 4 = 0: Select voltage measurement
  - Bit 4 = 1: Select current measurement
- If AGC is enabled, the analog input lets you measure input signals in the range of 0 ... 40 V.

  If AGC is disabled, the analog input lets you measure input signals in the range of 0 ... 5 V.
- If current measurement is enabled, the analog input lets you measure current signals of 0 or 4 ... 20 mA.

  Effectively, this mode changes the input impedance of the analog input to 240 Ω. 20 mA generate a 4.8 V signal, 4 mA generate a 960 mV signal, and 0 mA result in a 0 V signal.
- To allow for backwards compatibility, writing a value of **0x81** to sub-index 2 will disable the AGC mode. The value of **0x90** will disable the current measurement mode to enable voltage measurement.

| | |
|---|---|
| **Sub-index 4** | The function of sub-index 4 is described below:<br>■ Sub-index 4 lets you read the value of the latest measured analog input signal.<br>■ With AGC enabled, the measured value will range between 0 ... 8,191.<br>■ With AGC disabled, the measured value will range between 0 ... 1,023. |
| **Sub-index 5** | The function of sub-index 5 is described below:<br>■ Sub-index 5 lets you read the value of the latest measured analog input signal, too.<br>■ The value is indicated in millivolts (mV).<br>■ With AGC enabled, the measured value will range between 0 ... 40,000.<br>■ With AGC disabled, the measured value will range between 0 ... 5,000. |
| **Sub-index 6** | The function of sub-index 6 is described below:<br>■ Sub-index 6 lets you read the maximum value that can be output via sub-index 4. |
| **Sub-index 7** | The function of sub-index 7 is described below:<br>■ Sub-index 7 lets you read the maximum value that can be output via sub-index 5. |
| **Current mode** | If this input is used as current input (0 ... 20 mA), it measures the voltage through an internal resistor of 240 Ω. That is why a current of 20 mA results in a voltage of 4.8 V. This value corresponds to a measured value of 982. The full-scale value of 1023 is reached if the input current is 20.8 mA. |

# Object "Voltage Sense Analog Input" (Index 0x2210)

**Voltage Sense Analog Input (Index 0x2210)**

This read-only object returns as process value the measured value of the three power feeds.

The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2210 | 0 | 6 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Not used | |
| | 3 | 0 | Not used | |
| | 4 | 0 | Process value 0:<br>STANDARD FEED voltage [mV] | ro |
| | 5 | 0 | Process value 1:<br>IGNITION FEED voltage [mV] | ro |
| | 6 | 0 | Process value 2:<br>PROTECTED FEED ON/OFF | ro |

**Sub-index 4**

The function of sub-index 4 is described below:

- Sub-index 4 lets you read the measured voltage of STANDARD FEED in millivolts.

**Sub-index 5**

The function of sub-index 5 is described below:

- Sub-index 5 lets you read the measured voltage of IGNITION FEED in millivolts.

**Sub-index 6**

The function of sub-index 6 is described below:

- Sub-Index 6 lets you read out whether PROTECTED FEED (after the safety switch/relay) is enabled or disabled:
  - 0: PROTECTED FEED disabled
  - 1: PROTECTED FEED enabled
- This object does not have an analog measurement.

## Object "Feed Currents" (Index 0x2211)

**Feed Currents (Index 0x2211)**

The object **Feed Currents** is read-only. This object lets you read out the latest measurements of the STANDARD FEED and PROTECTED FEED currents. The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2211 | 0 | 2 | Number of entries | ro (read only) |
| | 1 | 0 | Current measurement STANDARD FEED | ro |
| | 2 | 0 | Current measurement PROTECTED FEED | ro |

**Sub-index 1**

The function of sub-index 1 is described below:

- Sub-index 1 lets you read the measured current of STANDARD FEED in mA.

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 lets you read the measured current of PROTECTED FEED in mA.

# Object "Analog Output" (Index 0x2300)

**Analog Output (Index 0x2300)**

This object lets you configure the analog output. Also, the analog output voltage/current can be set as process value.

The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2300 | 0 | 5 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Instruction | rw (read & write) |
| | 3 | 0x05 | Not used | |
| | 4 | 0 | Process value 0: Output voltage | rw |
| | 5 | 0 | Process value 1: Output current | rw |

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 lets you select between the following modes of operation.
  - 0x00: Disabled, no output function
  - 0x01: Constant output current
  - 0x02: Constant output current (ratiometric value specified)
  - 0x03: Constant output current (absolute value specified)
- To select one of the above modes, enter the corresponding value into sub-index 2.
- When you read out sub-index 2, the currently set mode is returned.
  The following information can be obtained:
  - 0x00: Disabled, no output function
  - 0x01: Constant output current
  - 0x02: Constant output current (ratiometric value specified)
  - 0x03: Constant output current (absolute value specified)
  - 0x08: Short-circuit to ground fault has been detected

**Sub-index 4**

The function of sub-index 4 is described below:

- Sub-index 4 lets you output a specific analog output voltage.
- With mode "Constant output voltage (ratiometric value specified)" enabled, the value will range between 0 ... 1,023.
  This value range relates to 0 ... 100 % of the input voltage.
- With mode "Constant output voltage (absolute value specified)" enabled, the value specifies the output voltage in mV units.
  If you enter a value that is larger than the maximum value, the output voltage will be clipped.
- The maximum output voltage of the analog output will always be slightly less than STANDARD FEED voltage.
- Sub-index 4 lets you read out the recently measured output voltage in mV units.

**Sub-index 5**

The function of sub-index 5 is described below:

- Sub-index 5 lets you set the analog output current.
- With mode "Constant output current" enabled, the value specifies the output current in 1 mA units.
- With mode "Constant output voltage (absolute value specified)" enabled, the value specifies the desired maximum output current.

  If the specified output voltage causes the output current to exceed this value, the output is clipped to control the output current.
- The maximum output voltage of the analog output will always be slightly less than STANDARD FEED voltage.
- Sub-index 5 lets you read out the recently measured output current in 1 mA units.

# Objects "PWM Output" (Index 0x2400 ... 0x2402)

**PWM Output 1 ... 3 (Index 0x2400 ... 0x2402)**

These objects let you configure the three PWM outputs. Also, the controlled output current or a PWM duty cycle can be set as process value. The structure of these objects is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2400 ... 0x2402 | 0 | 14 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Operating mode | rw (read & write) |
| | 3 | 0 | Not used | |
| | 4 | 0 | Process value 0: Output current | rw |
| | 5 | 0 | Process value 1: Pulse control factor | rw |
| | 6 | 2,500 | Parameter 0: Maximum value | ro |
| | 7 | 1,023 | Parameter 1: Maximum value | ro |
| | 8 | 0 | Predictor parameter | rw |
| | 9 | 0 | Proportional parameter | rw |
| | 10 | 0 | Integrator parameter | rw |
| | 11 | 0 | PWM predictor auto-tune function | rw |
| | 12 | 0 | Saving the system parameter settings | rw |
| | 13 | 0 | Locking the controller | rw |
| | 14 | 0 | PWM, averaging window (x times 8 ms) of analog-to-digital converter (ADC) | rw |

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 lets you select one of the following modes of operation.
  - 0x01: Current-controlled PWM output
  - 0x02: PWM output with static duty cycle
- To select one of the above modes, enter the corresponding value into sub-index 2.
- In static PWM duty-cycle output mode the output current will not be controlled.

  However, it will be monitored. If the measured current exceeds a user set threshold, the PWM output is disabled and the JXM-IO-E02 reports an error. The maximum value is to be entered into sub-index 6.

**Sub-index 4**

The function of sub-index 4 is described below:

- In current-controlled PWM output mode, write to sub-index 4 to set the output current.
- The value is in the range of 0 ... 2499 mA.
- Sub-index 4 lets you read out the recently measured output current in 1 mA units.

**Sub-index 5**

The function of sub-index 5 is described below:

- Sub-index 5 is used to set the PWM duty cycle.
- The value must be in the range of 0 ... 1,023, where 0 is a 0 % duty cycle and 1,023 is a 100 % duty cycle.
- Use the mode "PWM Output with Static Duty Cycle" in order to use the PWM output as a digital output.
- Reading sub-index 5 returns the most recent PWM duty cycle as a value in the range 0..1,023.

**Sub-index 6**

The function of sub-index 6 is described below:

- Sub-index 6 lets you read out the maximum value that can be input via sub-index 4.

**Sub-index 7**

The function of sub-index 7 is described below:

- Sub-index 7 lets you read out the maximum value that can be input via sub-index 5.

**Sub-index 8**

The function of sub-index 8 is described below:

- In "Current-Controlled PWM Output" mode sub-index 8 lets you set the Predictor parameter for the current control algorithm.
- This parameter is an unsigned 16-bit word where the least significant byte is the divisor and the most significant byte is the multiplier.
- The least significant byte of this parameter is not allowed to be zero because it is a divisor.

**Sub-index 9**

The function of sub-index 9 is described below:

- In "Current-Controlled PWM Output" mode sub-index 9 lets you set the Proportional parameter for the current control algorithm.
- This parameter is an unsigned 16-bit word where the least significant byte is the divisor and the most significant byte is the multiplier.
- The least significant byte of this parameter is not allowed to be zero because it is a divisor.

**Sub-index 10**

The function of sub-index 10 is described below:

- In "Current-Controlled PWM Output" mode sub-index 10 lets you set the Integrator parameter for the current control algorithm.
- This parameter is an unsigned 16-bit word where the least significant byte is the divisor and the most significant byte is the multiplier.
- The least significant byte of this parameter is not allowed to be zero because it is a divisor.

**Current control**

In "Current-Controlled PWM Output" mode the PWM duty cycle is controlled using the above three parameters in the following formula:

$$PWM_{DutyCycle} = \frac{Pre_{mul} \cdot Current_{Demand}}{Pre_{div}} + \frac{Pro_{mul} \cdot Error}{Pro_{div}} + \frac{Int_{mul}}{Int_{div} \cdot IntegratedError}$$

Where:

- $Pre_{mul}$ and $Pre_{div}$ are the Predictor multiplication and division factors (sub-index 8),
- $Pro_{mul}$ and $Pro_{div}$ are the Proportional multiplication and division factors (sub-index 9),
- $Int_{mul}$ and $Int_{div}$ are the Integrator multiplication and division factors (sub-index 10),
- $Current_{Demand}$ is the user input in milliamp,
- Error is the difference between the measured and commanded output current (also in milliamp).
- Integrated Error is the integral of the error signal.

**Calculating the PWM duty cycle**

When a new output current is requested, the "Error" and "Integrated Error" terms are zero. The output duty cycle is therefore calculated based on the user input and the Predictor parameters. To ensure that this first output level is accurate, the predictor parameter must be set for the load that it will be driving.

After the initial duty cycle calculation, the PWM algorithm uses the difference between the measured output current and the current demand to adjust the PWM duty cycle. The proportional and integrator parameters influence how fast the algorithm responds to a difference between the measured and demanded current. These two parameters also determine how much overshoot there will be.

All three parameters are highly dependent on the load that is being driven. Therefore it is the users' responsibility to tune these parameters for their own application.

If the PWM is already driving an output at a certain current level and a new output current is requested, the algorithm uses the actual output current to calculate the new duty cycle. This method reduces the sensitivity to incorrect predictor parameters, but does not remove it – these parameters will still affect the normal operation.

The PWM duty cycle is calculated as follows:

$$DC_0 = 1024 \times demand \times \frac{CR}{U_{batt}} \times 0.75$$

$$DC_n = DC_{n-1} + Err \times \frac{EP}{1024} + Int \times \frac{IP}{1024}$$

Where:

- $DC_0$ is the initial duty cycle (the first duty cycle after the current value has been entered (*Current Demand*)).
- *Current Demand* is the current demand in mA,
- *CR* is the coil resistance that has been specified in the system parameters or calculated by the software.
- $U_{batt}$ is the measured battery voltage (STANDARD FEED).
- $DC_n$ is the next duty cycle.
- $DC_{n-1}$ is the current duty cycle.
- *Err* is the error in the load current (difference between demand and actual value) in mA.
- *EP* is the error parameter – specified in the system parameters.
- *Int* is the integral of the error signal (basically the sum of the error currents, but the integral is multiplied by 0.25 after every cycle).
- *IP* is the integrator parameter – specified in the system parameters.

When you update the software of a device that has been configured for using the old PWM algorithm, the following settings are used:

The coil resistance is calculated by the predictor parameter. The formula that is now used to calculate DC0 has been integrated into the predictor parameter.

The proportional parameter (error parameter) is set to 160. The integrator parameter is set to 420. These values have been experimentally selected and have proven to be safe for a wide range of solenoid valves.

**Sub-index 11**

The function of sub-index 11 is described below:

- Sub-index 11 offers an automatic tuning of the Predictor parameter.
- To activate this function, write a 16-bit current value to sub-index 11.
- The system will then attempt to drive this current on the PWM and calculate what the predictor parameters must be in order to accurately guess the PWM duty cycle when a new current is requested.
- While this function is still running, the predictor parameter in sub-index 8 will read as **0**.
- As soon as the function completes (this may take up to 10 seconds, but is usually faster), the calculated Predictor parameters are available for reading from sub-index 8.

  The PWM algorithm will also use these parameters immediately.
- However, the newly calculated parameters will not be written to non-volatile memory. When the JXM-IO-E02 is reset or power cycled, these parameters will fall back to the previous parameters stored in the non-volatile memory.

  It is up to the user to first test the new parameters, read them from sub-index 8, and then write it to the System Parameters for permanent storage.

| | |
|---|---|
| **No-Load Detection** | The function "No-Load Detection" is described below:<br>■ No-load detection can be activated for each PWM channel individually.<br>■ When "Current Control" mode is selected, no-load will be detected if the duty cycle reaches maximum and the load current remains below the specified threshold.<br>■ In the "Static PWM Duty Cycle Output" mode, no-load is detected whenever the duty cycle is non-zero and the load current is below the threshold.<br>■ The no-load threshold is set in object **System Parameter** (index 0x4556, sub-index 40). |
| **Sub-index 12** | The function of sub-index 12 is described below:<br>■ To store the values from sub-index 11 to the system parameters, enter these values into sub-index 12. Then, enter "1" into 0x4556, sub-index 0 (enable write access to system parameters). |
| **Sub-index 13** | The function of sub-index 13 is described below:<br>■ When writing values to sub-index 12 is completed, sub-index 13 introduces a new parameter: **Control Inhibit**.<br>■ This parameter is used to slow down the PWM algorithm.<br>■ **Control Inhibit** is necessary because the PWM algorithm can run faster than the current rise/fall time in the load. |
| **Sub-index 14** | The function of sub-index 14 is described below:<br>■ For the purpose of current control the analog-to-digital converter measures via a shunt resistor the current that is produced by the PWM signal.<br>■ Sub-index 14 lets you set the number of measured values to be used by the ADC for averaging (x times 8 ms). |

## Object "H-Bridge" (index 0x2500)

**H-Bridge (Index 0x2500)**

This object lets you configure the H-Bridge. Also, this object can be used to read the output states. It returns the PWM duty cycle when the H-Bridge is in a PWM-controlled mode. The structure of the object "H-Bridge" is shown in the following table.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2500 | 0 | 7 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Operating mode | rw (read & write) |
| | 3 | 0 | Not used | |
| | 4 | 0 | Process value 0: Measured current | ro |
| | 5 | 0 | Process value 1: Output states/duty cycle | rw |
| | 6 | 1,023 | Parameter 0: Maximum output value | ro |
| | 7 | 7 | Parameter 1: Bridge configuration | ro |

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 lets you select between the following modes of operation.
  - 0x01: The two output channels (pins 69 and 70) are used as independent digital outputs.
  - 0x02: The output connected to pin 69 is a PWM-controlled high-side output, whereas the output connected to pin 70 is always low.
  - 0x04: The output connected to pin 70 is a PWM-controlled high-side output, whereas the output connected to pin 69 is always low.
- To select one of the above modes, enter the corresponding value into sub-index 2.

**Sub-index 4**

The function of sub-index 4 is described below:

- Sub-index 4 lets you read out the recently measured current in mA units.
- Note that current measurement is not available when the H-Bridge outputs are used as independent digital outputs.

**Sub-index 5**

The function of sub-index 5 is described below:

- When the H-Bridge outputs are used as two independent digital outputs, the least significant byte sets the output state:
  - Bit 1, 0: Pin 69 is set as output
  - Bit 5, 4: Pin 70 is set as output
- In the configuration as two independent digital outputs, the following data values are possible:
  - 0b00: Tri-state output

| | |
|---|---|
| | • 0b01: Output state OFF<br>• 0b10: Output state ON<br>▪ In PWM-controlled mode, a value in the range of 0 ... 1023 sets the PWM duty cycle.<br>▪ In PWM-controlled mode, sub-index 5 lets you set the PWM duty cycle. |
| **Sub-index 6** | The function of sub-index 6 is described below:<br>▪ Sub-index 6 lets you read out the maximum value for the duty cycle that can be input via Sub-Index 5. |

## Objects "Frequency Input" (Index 0x2600 ... 0x2601)

**Frequency Input (Index 0x2600 ... 0x2601)**

This object lets you configure input pins 54 and 55 either as frequency inputs or as simple digital inputs. In frequency input mode, the period length of the incoming signal is measured. The structure of these objects is shown in the table below.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x2600 ... 0x2601 | 0 | 7 | Number of entries | ro (read only) |
| | 1 | 0 | Not used | |
| | 2 | 0 | Operating mode | rw (read & write) |
| | 3 | 1 | Not used | |
| | 4 | 0 | Process value 0: Period length [ns] | ro |
| | 5 | 0 | Process value 1: Digital input state | ro |
| | 6 | 0xFFFFFFFF | Parameter 0: Maximum frequency value | ro |
| | 7 | 0 | Pulse count | ro |

**Sub-index 2**

The function of sub-index 2 is described below:

- To select the operating mode, enter the following values into sub-index 2:
  - 0: Frequency input mode (no pull-up/pull-down resistor)
  - 1: Digital input (active-low)
  - 2: Digital input (active-high)
  - 3: Frequency input (with pull-up resistor)
  - 4: Frequency input (with pull-down resistor)
  - **As of device revision 12.xx** a new parameter is available:

    5: Frequency input (with low operating point and hysteresis for **Variable Reluctance Sensor**)
- Sub-index 2 lets you read out the current operating mode.

**Sub-index 4**

The function of sub-index 4 is described below:

- In frequency input mode, sub-index 4 lets you read out the value of the latest measured period length.
- The returned value is a 32-bit unsigned integer specifying the period length of the signal in nanoseconds.
- This result is updated every 17 or 18 cycles of the external signal.

| | |
|---|---|
| **Sub-index 5** | The function of sub-index 5 is described below:<br>▪ In digital input mode, sub-index 5 lets you read out the current state of input pin 54 or 55.<br>▪ The following values are allowed:<br>• 0: Input state OFF<br>• 1: Input state ON |
| **Sub-index 7** | The function of sub-index 7 is described below:<br>▪ The two Frequency Input circuits will always count pulses on these inputs regardless of their input mode. The pulse period must not be less than 1 millisecond. The pulse must be active for at least 1 millisecond each period in order to be counted.<br>▪ The pulse count always starts at zero at power on/reset. The value is also reset to zero every time that it is read via this SDO.<br>▪ The returned value is a 32-bit unsigned integer.<br>▪ The frequency of the input signal must be less than 500 Hz. |

## Object "5 V Output" (0x2800)

**5 V Output (Index 0x2800)**

The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description |
|---|---|---|---|
| 0x2800 | 0 | 2 | Number of entries |
| | 1 | 0 | 5 V output error state |
| | 2 | 0 | 5 V output is activated |

**Sub-index 1**

The function of sub-index 1 is described below:

- Sub-index 1 returns the error state of the 5 V output.
  - Value **0** means no error.
  - Value **1** indicates that the protection has been activated due to an external fault.

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 returns the current drive status of the 5 V output.
  - 0: 5 V output is disabled
  - 1: 5 V output is enabled
- Sub-index 2 also supports write access (OS version 3.03.0.00 or higher). This lets you enable/disable the 5 V output for production testing.
  - Value **0** disables the 5 V output.
  - Value **1** enables the 5 V output.
- Only sub-index 2 is writable.

# Objekte "OS Update" (Index 0x4554) und EDS (Index 0x4555)

**OS Update (Index 0x4554)**

This object is used for OS updates.
It should not be accessed directly in most applications.

**Electronic Data Sheet (Index 0x4555)**

This object is read-only. This object lets you retrieve, for instance, the PCB revision. If you wish to contact the hotline of Jetter AG in case of a problem, please have this information ready.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x4555 | 0 | 15 | Number of entries | ro (read only) |
| | 1 | 0 | Status | ro |
| | 2 | 0 | Instruction | ro |
| | 3 | | Page 0: Version | ro |
| | 4 | | Page 0: Module code | ro |
| | 5 | | Page 0: Module name (string) | ro |
| | 6 | | Page 0: PCB revision | ro |
| | 7 | | Page 0: PCB options | ro |
| | 8 | | Page 1: Revision | ro |
| | 9 | | Page 1: Module serial number (string) | ro |
| | 10 | | Page 1: Production date: Day | ro |
| | 11 | | Page 1: Production date: Month | ro |
| | 12 | | Page 1: Production date: Year | ro |
| | 13 | | Page 1: Test device number | ro |
| | 14 | | Page 1: Test device version | ro |
| | 15 | | Page 0: Minimum OS version | ro |

## Object "System Parameter" (Index 0x4556)

**System Parameters (Index 0x4556)**

Use the object "System Parameters" to permanently change the parameters mentioned below. Any changes made to these parameters are stored in non-volatile memory and are therefore recovered when the JXM-IO-E02 is next powered up. The device must be rebooted for the changes to take effect.

Note that some of these settings can also be set using other SDO objects. However, the **System Parameters** object is the only way to make these changes permanently.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x4556 | 0 | 83 | Number of entries | ro (read only) |
| | 1 | 0 | Version | ro |
| | 2 | 0 | CAN bus termination | rw (read & write) |
| | 3 | 1 | CAN baud rate | rw |
| | 4 | 0x10 | CANopen® node ID | rw |
| | 5 | 1,000 | CANopen® heartbeat time period | rw |
| | 6 | 0x0A16 | PWM 1: Predictor parameter | rw |
| | 7 | 0x0302 | PWM 1: Proportional parameter | rw |
| | 8 | 0x0101 | PWM 1: Integrator parameter | rw |
| | 9 | 0x0A16 | PWM 2: Predictor parameter | rw |
| | 10 | 0x0302 | PWM 2: Proportional parameter | rw |
| | 11 | 0x0101 | PWM 2: Integrator parameter | rw |
| | 12 | 0x0A16 | PWM 3: Predictor parameter | rw |
| | 13 | 0x0302 | PWM 3: Proportional parameter | rw |
| | 14 | 0x0101 | PWM 3: Integrator parameter | rw |
| | 15 | 0 | Analog input 1: Mode select | rw |
| | 16 | 0 | Analog input 2: Mode select | rw |
| | 17 | 0 | Analog input 3: Mode select | rw |
| | 18 | 0 | Analog input 4: Mode select | rw |
| | 19 | 25 | Digital output 1 (STANDARD): Current limit | rw |
| | 20 | 25 | Digital output 2 (STANDARD): Current limit | rw |
| | 21 | 25 | Digital output 3 (STANDARD): Current limit | rw |
| | 22 | 25 | Digital output 4 (STANDARD): Current limit | rw |

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| | 23 | 25 | Digital output 5 (STANDARD): Current limit | rw |
| | 24 | 25 | Digital output 6 (STANDARD): Current limit | rw |
| | 25 | 25 | Digital output 7 (STANDARD): Current limit | rw |
| | 26 | 25 | Digital output 8 (STANDARD): Current limit | rw |
| | 27 | 50 | Digital output 9 (PROTECTED): Current limit | rw |
| | 28 | 50 | Digital output 10 (PROTECTED): Current limit | rw |
| | 29 | 50 | Digital output 11 (PROTECTED): Current limit | rw |
| | 30 | 50 | Digital output 12 (PROTECTED): Current limit | rw |
| | 31 | 50 | Digital output 13 (PROTECTED): Current limit | rw |
| | 32 | 50 | Digital output 14 (PROTECTED): Current limit | rw |
| | 33 | 50 | Digital output 15 (PROTECTED): Current limit | rw |
| | 34 | 50 | Digital output 16 (PROTECTED): Current limit | rw |
| | 35 | 1 | PWM output 1: Mode | rw |
| | 36 | 1 | PWM output 2: Mode | rw |
| | 37 | 1 | PWM output 3: Mode | rw |
| | 38 | 1 | Tri-state coding enable | rw |
| | 39 | 100 | Digital output: No-load threshold | rw |
| | 40 | 100 | PWM output: No-load threshold | rw |
| | 41 | 0 | Frequency input 1: Mode | rw |
| | 42 | 0 | Frequency input 2: Mode | rw |
| | 43 | 2,500 | H-bridge: Current limit | rw |
| | 44 | 100 | H-bridge: No-load threshold value | rw |
| | 45 | 2,500 | PWM output: Current limit | rw |
| | 46 | 0 | Digital inputs IN 1 ... IN 5: Active-low/active-high selection | rw |
| | 47 | 0 | Switch feed output: Initial output state | rw |

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| | 48 | 0 | Digital outputs: No-load detection enable | rw |
| | 49 | 0 | PWM: No-load detection enable | rw |
| | 50 | 0 | H-bridge: No-load detection enable | rw |
| | 51 | 0x00000C00 | Event-based PDO TX enable | rw |
| | 52 | 0xC000 | Digital output: Start-up current pulse select | rw |
| | 53 | 0 | PWM software mode | rw |
| | 54 | 10 | PWM 1: Control inhibit | rw |
| | 55 | 10 | PWM 2: Control inhibit | rw |
| | 56 | 10 | PWM 3: Control inhibit | rw |
| | 57 | 0 | H-bridge: Initial output mode | rw |
| | 58 | 100 | Analog output: Maximum output current | rw |
| | 59 | 2000 | Frequency input 1: High frequency reject | rw |
| | 60 | 2000 | Frequency input 2: High frequency reject | rw |
| | 61 | 2 | Frequency input 1: Number of periods | rw |
| | 62 | 2 | Frequency input 2: Number of periods | rw |
| | 63 | 0 | RX PDO timeout period | rw |
| | 64 | 0 | RX PDO timeout enable | rw |
| | 65 | 0 | Analog output: Initial mode | rw |
| | 66 | 128 | Frequency input 1: Timeout period | rw |
| | 67 | 128 | Frequency input 2: Timeout period | rw |
| | 68 | 15 | Analog input 1: Change threshold | rw |
| | 69 | 15 | Analog input 2: Change threshold | rw |
| | 70 | 15 | Analog input 3: Change threshold | rw |
| | 71 | 15 | Analog input 4: Change threshold | rw |
| | 72 | 2 | TX-PDO-1 - Minimum time | rw |
| | 73 | 2 | TX-PDO-2 - Minimum time | rw |
| | 74 | 2 | TX-PDO-3 - Minimum time | rw |
| | 75 | 2 | TX-PDO-4 - Minimum time | rw |

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| | 76 | 60 | TX-PDO-1 - Maximum time | rw |
| | 77 | 60 | TX-PDO-2 - Maximum time | rw |
| | 78 | 60 | TX-PDO-3 - Maximum time | rw |
| | 79 | 60 | TX-PDO-4 - Maximum time | rw |
| | 80 | 1 | PWM 1: Averaging window (x times 8 ms) of ADC | rw |
| | 81 | 1 | PWM 2: Averaging window (x times 8 ms) of ADC | rw |
| | 82 | 1 | PWM 3: Averaging window (x times 8 ms) of ADC | rw |
| | 83 | 0 | PWM frequency | rw |

**Version (sub-index 1)**

This parameter is read-only. The return value of this parameter should be **0**.

**CAN bus termination**

This parameter selects whether the CAN termination resistors (120 Ω) inside the JXM-IO-E02 must be activated (one each at both ends of the CAN bus).

The following values are allowed:

- 0x00: Both resistors are disabled
- 0x01: Resistor at the end of the CAN bus is enabled (default value)
- 0x02: Resistor at the beginning of the CAN bus is enabled
- 0x03: Both resistors are enabled

**When using the built-in CAN termination resistors, keep the following fact in mind:**

If a device with enabled CAN termination resistors fails, the whole CAN communication breaks down.

**CAN baud rate**

This parameter lets you select the CAN Baud rate.

The following values are allowed:

- 0: 125 kBaud
- 1: 250 kBaud (default)
- 2: 500 kBaud
- 3: 1 MBaud

**CANopen® node ID**

Even with coding via tri-state inputs 1 (pin 67) and 2 (pin 68) enabled, this parameter lets you set the node ID of the JXM-IO-E02.

If the device is configured NOT to use the tri-state inputs 1 and 2 for selecting the node ID, then the value stored in this parameter will be the final node ID.

The following values are allowed:

- The value is in the range of 0x01 and 0x76.
- The default value is 0x10.

**CANopen® heartbeat time period**

This is the time period, specified in milliseconds (ms), at which the JXM-IO-E02 will transmit a CANopen® heartbeat message. The own heartbeat status is sent as content of this message.

Time periods less than 250 ms are allowed by CANopen® but do not make practical sense for the JXM-IO-E02 and are therefore not allowed.

The following values are allowed:

- The legal range for values is between 250 and 65,535 ms.
- The default value is 1000.

**PWM - Predictor, proportional and integrator parameters**

Please refer to the description on object 0x2400, sub-index 8 ... 10.

**Analog input mode selection**

The parameters of **sub-index 15 ... 18** let you select the initial mode for the analog inputs at power on. These parameters correspond to objects **0x2200 ... 0x2203** (sub-index 2).

For mode "current measurement" the following values with their predefined value ranges are allowed:

- 0: 0 ... 5 V (default)
- 1: 0 ... 40 V
- 4: 0 ... 5 V with the pull-up resistor 240 Ω enabled.
- 5: 0 ... 40 V with the pull-up resistor 240 Ω enabled.
- 16: 0 ... 20 mA with the pull-up resistor 240 Ω enabled.
- 20: 0 ... 5 V with the pull-up resistor 240 Ω enabled.
- All other values are illegal.

**Digital outputs 1 ... 8 (STANDARD) - Current limit**

The parameters of **sub-index 19 ... 26** let you set the current limits for the digital outputs (STANDARD FEED).

The data is in units of 100 mA, i.e. 1 = 100 mA; 25 = 2.5 A.

The following values are allowed:

- The current limit must be in the range 1 ... 30 (100 mA ... 3 A).
- The default value is 25 (2.5 A).

**Digital outputs 9 ... 16 (PROTECTED) - Current limit**

The parameters of **sub-index 27 ... 34** let you set the current limits for the digital outputs (PROTECTED FEED).

The data is in units of 100 mA, i.e. 1 = 100 mA; 25 = 2.5 A.

The following values are allowed:

- The current limit must be in the range 1 ... 55 (100 mA ... 5.5 A).
- The default value is 50 (5 A).

**Digital outputs 1 ... 16 - Current measurement**

The current measurement of the JXM-IO-E02 is temperature dependent. At low temperatures the output current will be slightly larger than the limit above before being limited and at high temperatures the output current will be slightly smaller.

The following formula gives the relation between the specified current and the actual measured current:

$$I_{Ist} = I_{Soll} * \frac{K}{9.500}$$

Where K is taken from the following table:

| Load current | K at T = -40 °C | K at T = 25 °C | K at T = 125 °C |
|---|---|---|---|
| 0.5 A | 12,000 | 12,000 | 12,000 |
| 2.5 A | 10,000 | 9,700 | 9,300 |
| 5.0 A | 10,000 | 9,700 | 9,300 |

The temperature specified in the above table is not the ambient temperature, but rather an internal device temperature.

This temperature will be at least 20°C higher than the ambient when the JXM-IO-E02 has been working for a few minutes.

Especially for lower output currents, the measurement value will be far less accurate. It is recommend for setting the no-load and overcurrent thresholds that the thresholds are set with healthy safety margins.

**PWM outputs - Mode selection**

The parameters of **sub-index 35 ... 37** let you select the initial mode for the PWM outputs 1 ... 3 at power on. Please refer to the description on object **0x2400**, sub-index 2.

The following values are allowed:

- 0: Output is disabled.
- 1: Current-controlled operation (default).
- 2: Static duty cycle operation (default).

**Tri-state coding enable**

When this parameter is set to 1, the JXM-IO-E02 will use the tri-state inputs to calculate its node ID.

Set this parameter to "0" in order to disable this function.

The default value is "1".

**Digital outputs: Threshold for no-load detection**

**Sub-index 39** lets you set the no-load detection threshold for the 16 digital outputs.

The threshold is specified in milliamps. The following values are allowed:

- The allowed range is 50 ... 250 (50 mA ... 250 mA).
- The default value is 100 (100 mA).

When a digital output's load current is low (less than 1 A), the current measurement becomes inaccurate (refer to section "Digital outputs 1 ... 16 - Current measurement"). If a threshold current of 100 mA is specified, it is likely that the actual measured current is 126 mA.

**PWM outputs: Threshold for no-load detection**

**Sub-index 40** lets you set the no-load detection threshold for the 3 PWM outputs. No-load detection is available only in "static PWM duty-cycle output mode" (no current control).

The threshold is specified in milliamps. The following values are allowed:

- The allowed range is 10 ... 1,000 (10 mA ... 1,000 mA).
- The default value is 100 (100 mA).

**Frequency input - Mode selection**

**Sub-index 41 ... 42** let you select the initial mode for the frequency inputs 1 ... 2 at power on. Please refer to the description on objects **0x2600 ... 0x2601**, sub-index 2.

The following values are allowed:

- 0: Mode "Frequency measurement without bias" (default)
- 1: Mode "Digital input with pull-up resistor"
- 2: Mode "Digital input with pull-down resistor"
- 3: Mode "Frequency measurement with pull-up resistor"
- 4: Mode "Frequency measurement with pull-down resistor"

**H-bridge: Current limit**

**Sub-index 43** lets you set the output current limit of the H-bridge.

The data is in units of 1 mA (i.e. 1 = 1 mA; 2,500 = 2.5 A).

The following values are allowed:

- The current limit must be in the range 250 ... 3,000 (250 mA ... 3.0 A).
- The default value is 2,500 (2.5 A).

**H-bridge: Threshold for no-load detection**

**Sub-index 44** lets you set the no-load detection threshold for the H-bridge.

The threshold is specified in milliamps. The following values are allowed:

- The allowed range is 100 ... 250 (100 mA ... 250 mA).
- The default value is 100 (100 mA).

**PWM output: Current limit**

**Sub-index 45** lets you set the output current limit of the PWM outputs 1 ... 3. This parameter applies only when the outputs are used in "static PWM duty-cycle output mode" without current control. The data is in units of 1 mA (i.e. 1 = 1 mA; 2,500 = 2.5 A).

The following values are allowed:

- The current limit must be in the range 500 ... 2,500 (500 mA ... 2.5 A).
- The default value is 2,500 (2.5 A).

**Digital inputs 1 ... 5: Bias value**

**Sub-index 46** lets you set the bias value of digital inputs 1 ... 5 at power on. Please refer to the description on object **0x2100**, sub-index 2.

Each digital input is represented by a single bit in the 5-bit word.

- Bit 0 = Digital input 1
- Bit 1 = Digital input 2
- ...
- Bit 4 = Digital input 5

Legal bit values are:

- 0: Pull-up resistor (active-low) enabled (default).
- 1: Pull-down resistor (active-high) enabled

**Switch feed output states**

**Sub-index 47** lets you set the inital state of the two switch feed outputs at power on. Please refer to the description on object **0x2103**, sub-index 5.

The following values are allowed:

- The value must be in the range of 0 ... 3.
- The default value is "0" selecting both outputs to be disabled.

**Digital outputs: No-load detection enable**

**Sub-index 48** lets you enable/disable the no-load detection for the 16 digital outputs.

Each channel is represented by a single bit in the 16-bit word.

- Bit 0: Channel 1 (OUT 1)
- Bit 1: Channel 2 (OUT 2)
- ...
- Bit 14: Channel 15 (OUT 15)
- Bit 15: Channel 16 (OUT 16)

To enable/disable no-load detection set the corresponding bit value: The following values are allowed:

- 0: No-load detection is disabled (default for all channels)
- 1: No-load detection is enabled

No validation is performed on this parameter because all possible values (0 ... 65535) are legal.

**PWM outputs: No-load detection enable**

**Sub-index 49** lets you enable/disable the no-load detection for the 3 PWM outputs.

Each PWM output is represented by a single bit:

- Bit 0: PWM output 1
- Bit 1: PWM output 2
- Bit 2: PWM output 3

To enable/disable no-load detection set the corresponding bit value. The following values are allowed:

- 0: No-load detection is disabled (default).
- 1: No-load detection is enabled

The legal range for values is 0 ... 7 (inclusive).

**H-bridge: No-load detection enable**

**Sub-index 50** lets you enable/disable the no-load detection for the H-bridge. This parameter applies only when the H-bridge is used in PWM mode.

The following values are allowed:

- 0: No-load detection is disabled (default).
- 1: No-load detection is enabled

**Event-based PDO TX enable**

**Sub-index 51** lets you select which inputs are allowed to trigger event-based PDO transmission. For more information, refer to section **CANopen® PDO Specification** (see page 108).

Each of the 21 inputs is represented by a single bit in the 32-bit word.

- Bit 0: Digital input IN 1
- Bit 1: Digital input IN 2
- Bit 2: Digital input IN 3
- Bit 3: Digital input IN 4
- Bit 4: Digital input IN 5
- Bit 16: Digital input IN 6
- Bit 17: Digital input IN 7
- ...

| | |
|---|---|
| | ▪ Bit 30: Digital input IN 20<br>▪ Bit 31: Digital input IN 21<br>To enable/disable event-triggered transmission of a PDO message set the corresponding bit value:<br>▪ 0: Event-triggered transmission is disabled<br>▪ 1: Event-triggered transmission is enabled<br>▪ The default value is "x00000C000" disabling event-triggered transmission of a PDO message for all inputs. This setting enables transmission of frequency measurement PDOs.<br>▪ Legal value range: The three reserved bits (bits 7, 6 and 5) must be zero. This means that the value of the least significant byte must never be higher than 31. |
| **Digital output - Startup current pulse load** | **Sub-index 51** lets identify digital output channels that may draw a large current pulse at startup (inrush current).<br>This parameter accepts a 16-bit integer value. Each bit corresponds to one output channel:<br>▪ Bit 0: Digital output 1<br>▪ ...<br>▪ Bit 15: Digital output 16<br>For outputs that drive higher loads, set the corresponding bit in this parameter to 1. This informs the software to switch on this channel separately from all other channels to minimize the effect of the in rush currents in the device<br>The default value is 0xC000.<br>▪ This setting enables this feature for output channels 15 and 16.<br>▪ Because all possible values (0 to 65535) are valid, no validation checking is done on this parameter. |
| **PWM software mode** | ▪ Refer to the PWM parameters below. |
| **PWM - Control inhibit** | **Sub-index 54 ... 56** lets you set the rate at which the PWM current control algorithm runs.<br>▪ The cycle time of this algorithm is 8 ms.<br>▪ This parameter sets the delay between every new control algorithm.<br>▪ The legal range for values is 1 ... 100 (inclusive).<br>▪ The default value is "10". |
| **H-bridge: Default mode of H-bridge output** | **Sub-index 57** lets you set the default mode of the H-bridge output.<br>▪ The value range is the same as for SDO object **0x2500**, sub-index 2.<br>▪ The following values are allowed:<br>• 0: H-bridge disabled (default)<br>• 1: Independent push/pull digital outputs<br>• 2: Static duty cycle PWM: left-side output high, right-side output low<br>• 4: Static duty cycle PWM: right-side output high, left-side output low |

**Analog output – Maximum current**

**Sub-index 58** lets you set the maximum output current [mA] for the analog output.

- The legal range for values is 1 ... 100.
- The default value is "100".

This parameter only has an effect when the analog output is used in one of the voltage output modes.

**Frequency inputs - High frequency reject**

**Sub-index 59 ... 60** lets you set the maximum frequency that should be measured by frequency inputs 1 ... 2.

This parameter allows high frequency noise signals to be ignored.

The following values are allowed:

- 0: To disable this function
- Value range 100 Hz ... 20.000 Hz
- The default value is "2000".

**Frequency input - Number of wave periods**

**Sub-index 61 ... 62** lets you set the number of signal periods counted and averaged for calculating the frequency for frequency inputs 1 ... 2.

The higher this number is, the more immune the measurement is to noise disturbances, but at the cost of taking longer to respond to changes in the signal frequency.

The following values are allowed:

- Value Range 1 ... 15.
- The default value is "2".

**Timeout for receiving PDO messages**

**Sub-index 63** lets you set the maximum time within which the JXM-IO-E02 must receive PDO commands before disabling outputs. This value is specified in milliseconds [ms].

The following values are allowed:

- 0: To disable this function (default)
- Value Range 500 ... 60,000 ms.

For more information, refer to the description of the parameter "PDO timeout enable".

**PDO timeout enable**

**Sub-index 64** lets you enable/disable PDO timeout (see sub-index 63) for each of the four RX-PDOs.

Each bit corresponds to a RX-PDO:

- Bit 0: RX-PDO-1
- Bit 1: RX-PDO-2
- Bit 2: RX-PDO-3
- Bit 3: RX-PDO-4

To enable/disable this feature set the corresponding bit value.

- 0: The feature is disabled (default).
- 1: The feature is enabled

The legal range for values is 0 ... 15 (inclusive).

**Analog output - Default output mode**

**Sub-index 65** lets you set the default mode of the analog output.

The value range is the same as for SDO object **0x2300**, sub-index 2. The following values are allowed:

- 0: Output disabled (default)
- 1: Current-controlled
- 2: Voltage-controlled (ratiometric)
- 3: Voltage-controlled (absolute)

**Frequency inputs 1 ... 2 - Timeout**

The parameters of **sub-index 66 ... 67** (frequency input 1 ... 2) let you set the maximum timeout period for the system to wait for a signal edge.

If no signal is received during this period, the frequency measurement aborts and returns the value **0**.

This parameter is specified in units of 16 ms. The following values are allowed:

- Value range 0 ... 128.
- The default value is "128".

**Analog Inputs 1 ... 4: Change threshold**

The parameters of **sub-index 66 ... 67** let you set the change threshold for the analog outputs 1 ... 4. The change threshold specifies the change in measurement that will register as an input change for triggering event-based PDO transmission.

The change in measurement must be at least 10 steps in order to trigger a PDO transmit.

The following values are allowed:

- Value range 1 ... 200.
  A value of 1 corresponds to 0.5 per cent. For example, 2 means 1 % of the full 1024-step measurement range.
- The default value is 15 which is 7.5 % of 1024 or 77 measurement steps.

**TX-PDO 1 ... 4: Minimum time between transmissions**

The parameters of **sub-index 72 ... 75** let you set the minimum amount of time between automatic (event-based) transmissions of TX-PDO-1.

If an input change triggers transmitting this PDO, the specified minimum amount of time must pass before the PDO will retransmit for a new input change event.

Requesting this PDO with CAN RTR messages will also reset this delay time.

This parameter is specified in units of 50 ms. The following values are allowed:

- Value range 0 ... 2540.
- Value **0** disables this feature.
- The default value is **2** which corresponds to a time period of 100 ms.

**TX-PDO 1 ... 4: Maximum time between transmissions**

The parameters of **sub-index 76 ... 79** let you set the maximum amount of time between automatic (event-based) transmissions of TX-PDO-1.

If no input change event occurs that triggers transmitting the PDO for the duration specified by this parameter, the device will transmit the PDO regardless. So, the external controller can have confidence that input signals are still being monitoring.

This parameter is specified in units of 50 ms. The following values are allowed:

| | |
|---|---|
| | ▪ Value range 0 ... 254.<br>▪ Value **0** disables this feature.<br>▪ The default value is 60 which corresponds to a time period of 3 ms. |
| **PWM 1 ... 3: Averaging window (x times 8 ms) of ADC** | The parameters of **sub-index 80 ... 82** let you select the ADC averaging window.<br>Every 8 seconds a current measurement takes place. This parameter returns the average value for the last x current measurement. Where x is the value that is specified in this parameter. The following values are possible:<br>• 1 (default): Displays the last current measurement value.<br>• 2 (default): Displays the average value of the last two current measurements.<br>• 3 (default): Displays the average value of the last three current measurements.<br>• ...<br>• 63 (default): Displays the average value of the last 63 current measurements.<br>• 64 (default): Displays the average value of the last 64 current measurements. |
| **PWM frequency - Values** | **Sub-index 83** lets you set the frequency for the 3 PWM outputs.<br>The following values are allowed:<br>• 0: 1.954 kHz (default)<br>• 1: 977 Hz<br>• 2: 488 Hz<br>• 3: 244 Hz<br>• 4: 122 Hz<br>• 5: 61 Hz<br>• 6: 30 Hz |

## Validating and setting system parameters

**Validating system parameter**

The first check is the system parameter version number. This value must not be 0xFF. If it is, this is considered to indicate an unprogrammed EEPROM. Therefore, all system parameters are set to their default values.

The next check counts the number of bytes in the system parameters that have a value of 0xFF. If the count finds more than 50 % at this value, all system parameters are invalidated and reset to their default values.

Individual parameters are range-checked at start-up. Each parameter has a specific minimum and maximum value or data pattern that must be adhered to. These ranges and/or patterns are described in the section above. If the value of any parameter does not match the requirements at start-up, the value of this parameter is reset to the default value.

If any of the three tests above finds a problem in the system parameters, EMCY 0x8002 is triggered.

Finally, the slave controller requests a copy of the system parameters as it is stored in the master controller. If the copy from the master controller does not match the slave, the master controller is reprogrammed to match the values stored in the slave. If this happens, the EMCY 0x8001 is triggered to indicate that a device reset occurred in order to reprogram and test system parameters.

**Writing/setting system parameters**

To write several system parameters in one go, proceed as follows:

| Step | Action |
|---|---|
| **1** | In object **0x4556**, sub-index 0, enter the value **0**. |
| **2** | Now, enter the required values into the individual sub-indeces 2 ... 83. |
| **3** | In object **0x4556**, sub-index 0, enter the value **1**. |
| **4** | Wait at least several 100 ms. |
| **5** | In object **0x4557** read out sub-index 1. Upon completion of the write process, the value **0** should be returned.<br>Note that during a write access it is possible that reading this sub-index could result in a timeout error. |

To make changes to only one system parameter, proceed as follows:

| Step | Action |
|---|---|
| **1** | Enter the new system parameter value. |
| **2** | In object **0x4557** read out sub-index 1. Upon completion of the write process, the value **0** should be returned.<br>Note that during a write access it is possible that reading this sub-index could result in a timeout error. |

**Setting the delay**

When writing to the system parameters, **make sure** that a `delay` is implemented after the SDO command.

The JXM-IO-E02 will reply to the SDO request to write to system parameters and will then write the data to non-volatile memory. This process can take as much as 50 ms. Therefore, it is recommended that a delay of 100 ms be implemented before the next SDO or PDO access to the same JXM-IO-E02.

**Setting PWM parameters**

Version 3 or higher of the JXM-IO-E02 software uses a different PWM current control algorithm. Please refer to the section *PWM output objects (Index 0x2400 ... 0x2402)* for information about the new PWM control algorithm.

As a result of this change, the control parameters had to change accordingly. But, for compatibility, the system parameters (index 0x4556) will allow control parameters based on both, the old and new algorithm. Therefore, if an older JXM-IO-E02 is upgraded from software version 2 to software version 3, the old PWM parameters will still work.

The system parameters at sub-indexes 6 ... 14 can stay exactly as before, or it can change to new values. The new parameter **PWM software mode** (sub-index 53) lets you select between old and new parameters. The default value for this parameter is to use "old" parameters. The three least significant bits of this parameter let you set the mode for each PWM output individually. Each bit corresponds to a PWM output:

- Bit 0: PWM 1
- Bit 1: PWM 2
- Bit 2: PWM 3

Legal bit values are:

- 0 = Old parameters
- 1 = New parameters

If the value **1** has been entered, the PWM system parameters change as listed below:

| Sub-index | **Description** | **Valid range and default** |
|---|---|---|
| 6, 9, 12 | Coil resistance for PWM 1, 2, and 3 | 250 ... 65278 (inclusive)<br>The default value is calculated during auto-tuning. |
| 7, 10, 13 | Error parameter for PWM 1, 2, and 3 | 0 ... 4095 (inclusive)<br>The default value is "160". |
| 8, 11, 14 | Integrator parameter for PWM 1, 2, and 3 | 0 ... 4095 (inclusive)<br>The default value is "420". |

If the value **0** has been entered, the following range checking applies:

| **Sub-index** | **Description** | **Valid range** |
|---|---|---|
| 6, 9, 12 | Predictor parameter (PWM-1, -2, and -3) | The value must be > 0, and the least significant byte must be >0. |
| 7, 10, 13 | Error parameter for PWM 1, 2, and 3 | The least significant byte must be > 0. |
| 8, 11, 14 | Integrator parameter for PWM 1, 2, and 3 | The least significant byte must be > 0. |

The value of parameter **PWM software mode** (sub-index 53) is also considered during the validation of the PWM parameters. If one or more parameters have an invalid value, all PWM parameters (including PWM software mode) are reset to default.

If "old parameters" are selected, the JXM-IO-E02 software will use the old-style predictor parameter (from sub-indexes 6, 9, and 12) to calculate an appropriate value for the coil resistance. The error parameter and integrator parameter are set to the default values of 160 and 420 respectively.

**Converting from old to new parameters**

The procedure for converting old parameters to new parameters is set out in section *PWM output objects (index 0x2400 ... 0x2402*. This procedure will make all the necessary changes to system parameters.

**Resetting system parameters to their default values**

The object **System Parameter** (index 0x4556) lets you reset system parameters to their default values. To prevent accidental triggering of this function, it is locked. To release this function, proceed as follows.

| Step | Action |
|---|---|
| **1** | First, write to **0x4556, sub-index 1** (version), the unsigned 32-bit value **0x01042006**. |
| **2** | Then, write the unsigned 32-bit value **0xC1EA5AFE** to the same sub-index. |
| **3** | Wait a few seconds. |
| ⇨ | When the JXM-IO-E02 receives the second code correctly, it will immediately reset the parameters in EEPROM. |

If a wrong code is entered at any stage, the JXM-IO-E02 will respond with a general CANopen® error message. Then you must repeat the whole procedure.

# Object "OS Status" (Index 0x4557)

**OS Status Object (Index 0x4557)**

The structure of this object is shown in the table below. This read-only object provides access to some useful information.

| Index | Sub-index | Default | Description |
|---|---|---|---|
| 0x4557 | 0 | 5 | Number of entries |
| | 1 | 0 | System parameters: Write status |
| | 2 | 0 | BSP state |
| | 3 | 2 | PCB revision number |
| | 4 | 0 | System parameters fault |
| | 5 | 0 | Not used |

**Sub-index 1**

The function of sub-index 1 is described below:

- Sub-index 1 returns the status of writing system parameter data to the EEPROM.
- As long as the writing process is still ongoing, this sub-index will return a non-zero value. 0 indicates that the process has finished.
- Note that during the writing to EEPROM it is possible that reading this sub-index could result in a CANopen® timeout error.

**Sub-index 2**

The function of sub-index 2 is described below:

- Sub-index 2 returns the BSP state machine state number.
- The BSP uses specific modes for certain tasks that must be performed at start-up. Once all these tasks have been completed successfully, the BSP returns state 255 which indicates normal operation.

**Sub-index 3**

The function of sub-index 3 is described below:

- Sub-index 3 returns the PCB revision number.
  - A value of **1** indicates the former hardware type.
  - A value of **2** indicates the latest hardware type.

**Sub-index 4**

The function of sub-index 4 is described below:

- Sub-index 4 returns either **0** or **1**. If during start-up one or more system parameters were set to a default value, this value is set to **1**.

## Object "Detailed Software Version" (Index 0x4559)

**Detailed Software Version (Index 0x4559)**

The structure of this object is shown in the table below. This read-only object returns the same software version as object **0x100A**, but in a 32-bit unsigned integer format which is compatible with the standard IP-type version numbers used at Jetter AG.

In addition, this object will also return the software version number for the two processors including their bootloader version numbers.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x4559 | 0 | 5 | Number of entries | ro (read only) |
| | 1 | - | Software version | ro |
| | 2 | 0 | Master OS version | ro |
| | 3 | 0 | Master bootloader version | ro |
| | 4 | 0 | Slave OS version | ro |
| | 5 | 0 | Slave bootloader version | ro |

# Object "Slave CAN Termination" (Index 0x4560)

**Slave CAN Termination Object (Index 0x4560)**

This object lets you read out or set the slave CAN bus (CANopen®) termination setting. The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description |
|---|---|---|---|
| 0x4560 | 0 | | Current termination resistor setting |

**Sub-index 0**

Sub-index 0 lets you read out the current setting, or make changes to it.

The following values are allowed:

- 0: Termination is disabled
- 1: Termination is enabled

Note that changing the CAN termination in this way is not permanent. This setting only has effect until the next time that the device restarts (power cycle, CANopen® reset command, writing to system parameters).

After a restart the JXM-IO-E02 again assumes the setting that is set in system parameters.

## Object "Master CAN Termination" (Index 0x4561)

**Master CAN Termination Object (Index 0x4561)**

This object lets you read out or set the master CAN bus termination setting. The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description |
|---|---|---|---|
| 0x4561 | 0 | | Current termination resistor setting |

**Sub-index 0**

Sub-index 0 lets you read out the current setting, or make changes to it.

The following values are allowed:

- 0: Termination is disabled
- 1: Termination is enabled

Note that changing the CAN termination in this way is not permanent. This setting only has effect until the next time that the device restarts (power cycle, CANopen® reset command, writing to system parameters).

After a restart the JXM-IO-E02 again assumes the setting that is set in system parameters.

# Object "Electronic Name Plate" (Index 0x4565)

**Electronic Name Plate Object (Index 0x4565)**

The structure of this object is shown in the table below.

| Index | Sub-index | Default | Description |
|---|---|---|---|
| 0x4565 | 0 | 5 | Number of entries |
| | 1 | 0 | Version number of the electronic name plate |
| | 2 | 0 | Instruction |
| | 3 | "" | Product serial number |
| | 4 | "" | Item number |
| | 5 | "" | Product revision |

**Sub-indexes 3, 4, 5**

The function of sub-indexes 3 ... 5 is described below.

Sub-indexes 3, 4, and 5 return strings containing the following information:

- Serial number of JXM-IO-E02 (32 characters)
- Item number (16 characters)
- Revision number of JXM-IO-E02 (16 characters)

## Object "User EEPROM Access" (Index 0x5000)

**User EEPROM Access (Index 0x5000)**

The structure of the object "User EEPROM Access" is shown in the following table. This object grants the user read/write access to the EEPROM.

| Index | Sub-index | Default | Description | Attributes |
|---|---|---|---|---|
| 0x5000 | 0 | 6 | Number of entries | ro (read only) |
| | 1 | 0 | Byte offset inside memory space | rw (read & write) |
| | 2 | 1,024 | Size of memory (in bytes) | ro |
| | 3 | 1 | Auto increment | ro |
| | 4 | - | Byte R/W access | rw |
| | 5 | - | 16-bit word R/W access | rw |
| | 6 | - | 32-bit word R/W access | rw |

**Sub-index 1**

The function of sub-index 1 is described below:

- To use this object, enter the byte offset inside the memory space in sub-index 1.
- If the byte offset is less than zero, the CANopen® error “Value of parameter written too low” is returned.
- If the byte offset is larger than the value in sub-index 2 (default value: 1,024), the CANopen® error “Value of parameter written too high” is returned.
- Also, if the byte offset is set to one of the last byte values and an attempt is made to read or write a 16-bit or 32-bit word which would cause reading/writing outside the memory space, the “General error” message is returned.
  Unfortunately CANopen® doesn’t have an error code that accurately describes this condition.
  Example:
  If the byte offset is 1,022 and an attempt is made to read a 32-bit word, this would normally try to read beyond the last memory address of 1023. This is not allowed and the message "General error" is returned.

**Sub-index 2**

The function of sub-index 2 is described below:

- The JXM-IO-E02 offers 1 kByte of EEPROM memory space. But for some special devices the amount may differ.
- Reading sub-index 2 returns the available memory size in bytes.
- This sub-index is read-only.

**Sub-index 3**

The function of sub-index 3 is described below:

- Use sub-index 3 to enable/disable the function "Auto Increment":
  - 0: Auto increment is disabled
  - 1: Auto increment is enabled
- Auto increment works as follows:
  - After either a read or write operation, the object will increment the offset in the memory space by the number of bytes that were transferred.
  - **Example:**

  After a byte read the byte offset is incremented by 1.
  After a 32-bit write the byte offset is incremented by 4.

**Sub-index 4**

The function of sub-index 4 is described below:

- Read sub-index 4 to read a byte from the memory.
- Enter a value into sub-index 4 to write a byte in the memory.

**Sub-index 5**

The function of sub-index 5 is described below:

- Read sub-index 5 to read a 16-bit word from the memory.
- Enter a value into sub-index 5 to write a 16-bit word in the memory.

**Sub-index 6**

The function of sub-index 6 is described below:

- Read sub-index 6 to read a 32-bit word from the memory.
- Enter a value into sub-index 6 to write a 32-bit word in the memory.

**Delay**

When writing to the EEPROM, a delay **must** be implemented after the SDO command. The JXM-IO-E02 first writes to the EEPROM memory. This may take a while before the JXM-IO-E02 transmits the SDO reply. This process can take at least 50 ms. Therefore, it is recommended that a delay of 100 ms be implemented before the next SDO or PDO access to the same JXM-IO-E02.

# 6.2 CANopen® PDO Specification

**Introduction**

This chapter describes the CANopen® PDO specification implemented on the JXM-IO-E02. PDO is short for Process Data Object. The PDO data allocation is fixed and cannot be changed by the application.

The JXM-IO-E02 allows PDO access when it has been set to operational state.

**Contents**


| Topic | Page |
|---|---|
| RX PDO allocation on the JXM-IO-E02 | 109 |
| TX PDO allocation on the JXM-IO-E02 | 110 |

# RX PDO allocation on the JXM-IO-E02

**PDO assignment and parameters**

The tables below show the allocation of RX PDOs implemented on the JXM-IO-E02. CANopen® objects are linked with their corresponding PDOs. Therefore, writing to that PDO will be the same as writing to that SDO index and sub-index.

The controller is able to read back the following data from the JXM-IO-E02 via PDO1_TX (0x200 + node ID):

| Byte offset | Index/sub-index | Size [byte] | Description |
|---|---|---|---|
| 0 | 0x2101/05 | 2 | Digital outputs |
| 2 | 0x2103/05 | 1 | Switch feed outputs |

The controller is able to read back the following data from the JXM-IO-E02 via PDO2_TX (0x300 + node ID):

| Byte offset | Index/sub-index | Size [byte] | Description |
|---|---|---|---|
| 0 | 0x2500/05 | 2 | H-bridge output state |
| 2 | 0x2400/04 | 2 | PWM-1 current |
| 4 | 0x2401/04 | 2 | PWM-2 current |
| 6 | 0x2402/04 | 2 | PWM-3 current |

Please note that specification of current values is allowed only in mode "Current-Controlled PWM Output".

If the PWM output is set to static duty-cycle mode, this parameter will actually change to sub-index 5 to allow writing to the duty-cycle register. The PDO interface can therefore be used to also select the duty cycle.

The controller is able to read back the following data from the JXM-IO-E02 via PDO3_TX (0x400 + node ID):

| Byte offset | Index/sub-index | Size [byte] | Description |
|---|---|---|---|
| 0 | 0x2300/04 | 2 | Analog output - voltage |
| 2 | 0x2300/05 | 2 | Analog output - current |

**New software**

The new JXM-IO-E02 software release (OS 3.00.0.00 or higher) contains a new system parameter feature. This feature lets the user detect a timeout for the a.m. RX-PDOs. When activated in the system parameters, the JXM-IO-E02 will expect to receive the RX PDOs in the specified time. If the RX PDO message does not arrive, the fault is triggered and all outputs controlled by this PDO are disabled.

## TX PDO allocation on the JXM-IO-E02

**PDO assignment and parameters**

The tables below show the allocation of TX PDOs implemented on the JXM-IO-E02. CANopen® objects are linked with their corresponding PDOs.

The controller is able to read back the following data from the JXM-IO-E02 via PDO1_RX (0x180 + node ID):

| Byte offset | Index/sub-index | Size [byte] | Description |
|---|---|---|---|
| 0 | 0x2101/04 | 2 | Digital outputs read back |
| 2 | 0x2100/04 | 2 | Digital inputs |
| 4 | 0x2100/04 | 1 | Tri-state input |
| 5 | - | 1 | Frequency input - Digital inputs |
| 6 | - | 1 | Switch feed output states |

The controller is able to read back the following data from the JXM-IO-E02 via PDO2_RX (0x280 + node ID):

| Byte offset | Index/sub-index | Size [byte] | Description |
|---|---|---|---|
| 0 | 0x2200/04 | 2 | Analog input # 1 |
| 2 | 0x2201/04 | 2 | Analog input # 2 |
| 4 | 0x2202/04 | 2 | Analog input # 3 |
| 6 | 0x2203/04 | 2 | Analog input # 4 |

The controller is able to read back the following data from the JXM-IO-E02 via PDO3_RX (0x380 + node ID):

| Byte offset | Index/sub-index | Size [byte] | Description |
|---|---|---|---|
| 0 | 0x2600/04 | 4 | Frequency input # 1 |
| 4 | 0x2601/04 | 4 | Frequency input # 2 |

For PDO-3_RX, the frequency input fields change to sub-index 5 of the respective objects when the frequency input is used as a digital input. This allows monitoring of the digital input level by means of PDO.

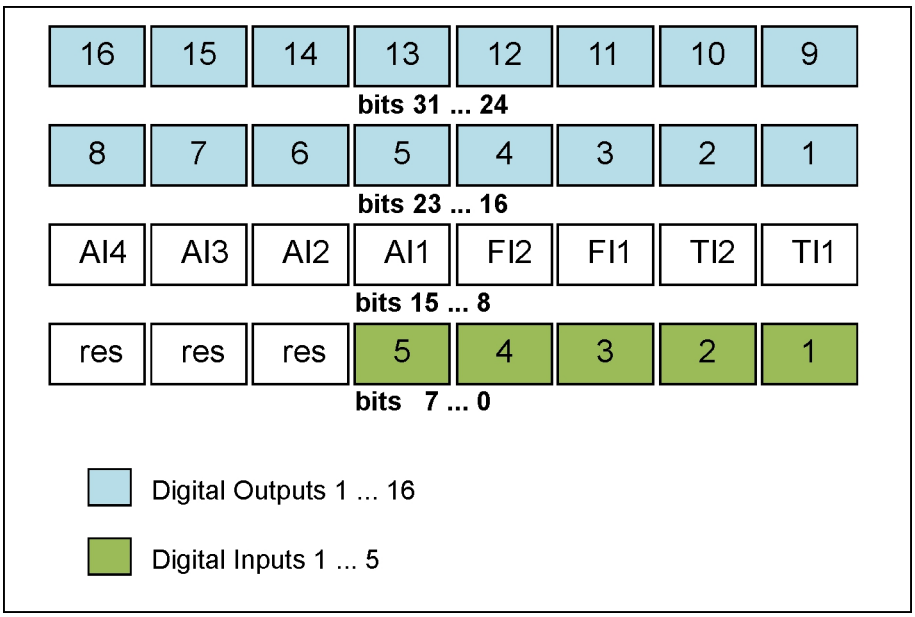
Normally, the PDOs are transmitted asynchronously on request. However, PDO1_RX can also be enabled to be transmitted asynchronously on events. This is done using the “Event-based PDO TX enable” option in the System Parameters interface.

Additionally, from OS version 2.10.0.01 PDO3_RX is also transmitted asynchronously on events. The event that triggers this is the completion of a frequency measurement. This function cannot be disabled and is only available for frequency measurement at this time.

TX PDOs are transmitted upon receiving a remote request (RTR) message. But, event-based triggering of the TX PDOs can also be enabled in the system parameters.

**Digital input - Event-based PDO transmission enable**

This parameter enables/disables triggering for individual inputs. The following diagram shows the bit mapping for this parameter:


16 15 14 13 12 11 10 9
bits 31 ... 24
8 7 6 5 4 3 2 1
bits 23 ... 16
AI4 AI3 AI2 AI1 FI2 FI1 TI2 TI1
bits 15 ... 8
res res res 5 4 3 2 1
bits 7 ... 0
Digital Outputs 1 ... 16
Digital Inputs 1 ... 5

In the diagram, the green bits represent the five digital input channels. The blue bits represent the sixteen digital output read back functions. The other bits are labelled:

- TI1 and TI2: Tristate inputs 1 and 2
- FI1 and FI2: Frequency inputs 1 and 2
- AI1, AI2, AI3, AI4: Analog inputs 1 through 4
- res: Reserved bits (must be set to zero)

For all digital inputs, the PDO is triggered when a state change is detected. In the case of the frequency inputs in "frequency measurement" mode, the PDO is triggered when a measurement is completed, but only if a minimum delay of 50 ms has expired. This is to prevent the PDO being transmitted every 1 ms, while the input signal has a frequency of 1 kHz or higher. Presently, configuring this delay is not possible yet.

A state change for the analog inputs is defined as a change in the measurement of at least 7.5 % (this percentage is now configurable in system parameters) of the full-scale measurement. This prevents triggering PDOs due to small noise signals. But it does mean that smaller changes in the analog value measured are not responded to. Therefore, it is recommended that especially the analog input PDO be requested by remote request (RTR) at regular intervals in addition to event-triggering.

**Time restrictions for automatic transmission of PDOs**

"Automatic transmission" means transmission of a PDO message which has not been requested by a CAN RTR message. Usually this implies that the PDO transmission has been triggered by an input change event.

Since software version 3.09.0.00, new system parameters have been added to configure time restrictions on these automatic transmissions:

1. Minimum cycle time

- A minimum period of time must pass between consecutive PDO messages transmitted. This is to prevent CAN bus overload when an input is changing very rapidly. Whenever a PDO message is transmitted, this delay timer is reset. This also applies the PDO message being transmitted due to a received CAN RTR message. The default delay time is set to 100 ms. This feature can be disabled for individual PDO channels.

2. Maximum cycle time

- A maximum period of time between PDO transmissions is also specified.
- Once this period has elapsed, a PDO with the current data is automatically transmitted if
  - no CAN RTR messages have been received
  - no input change event has triggered transmission of the PDO.
- This ensures that
  - the PDO data is always received by the external controller,
  - there is never a doubt whether the JXM-IO-E02 is still monitoring its inputs.

**Restrictions**

Due to design constraints some restrictions/limitations had to be applied to the CANopen® interface.

- SDO expedited transfer only supports 4-byte transfers. Any smaller data element must be extended to 32 bit before the SDO transfer.
- SDO segmented transfer is only supported on certain objects. Unless an object is documented to support segmented transfers, assume that it does not.
- SDO block transfer is NOT implemented.
- PDOs are not user configurable.
- PDOs are transmitted only asynchronous on request unless otherwise specified.
- Although emergency messages are transmitted to notify of detected faults, the CANopen® emergency handling system is not fully implemented.
- The Error Register does not save its state in non volatile memory. After each reset or power cycle, the error list is cleared.

# 7 Protection and diagnostic features - JXM-IO-E02

**Purpose of this chapter**

This chapter describes the available protection and diagnostics features implemented on the JXM-IO-E02. The following features are currently supported:

- Detecting an error by means of the application or visualization program
- Identifying the root cause of an error
- Troubleshooting an error that caused a fault message.

**Prerequisites**

To be able to troubleshoot a fault on the JXM-IO-E02, the following prerequisites must be fulfilled:

- The JXM-IO-E02 is either connected to a controller, or is integrated into the controller JCM-350-E03.
- The controller is connected to a PC.
- The JetSym programming software has been installed on the PC.
- The minimum requirements regarding modules, controllers and software are fulfilled.

**Background**

When a fault is detected, the JXM-IO-E02 will disable the function that caused the fault. It will transmit a CANopen® Emergency Object to inform the controller of the problem. The fault is also recorded in a history list of error events. These error events are compliant to the CANopen® "Pre-defined Error Field".

The external controller can immediately reactivate the function, but as long as the fault remains, the JXM-IO-E02 will again disable the function and retransmit the error notification.

**Locking-out outputs**

If an output has experienced a defined number of faults (short to ground, overcurrent, or overtemperature), the software will lock this output to prevent it being switched on again. If the controller demands switching on this output, the lock-out fault is triggered. Once an output is locked, the JXM-IO-E02 must be power cycled before the output can be activated again. Check the JXM-IO-E02 before you put it into normal operation.

**Contents**


| Topic | Page |
|---|---|
| Standard feed power input (STANDARD FEED) | 115 |
| Protected feed power input (PROTECTED FEED) | 116 |
| Digital Outputs 1 ... 8 (Standard Outputs) | 117 |
| Digital Outputs 9 ... 16 (Safety Outputs) | 118 |
| Analog output | 119 |
| PWM outputs 1 ... 3 | 120 |
| H-bridge | 121 |
| Switch feed outputs 1 ... 2 | 122 |
| Safety switch (solid-state relay) | 123 |
| 5 V reference output | 124 |
| Generic fault detection | 125 |

## Standard feed power input (STANDARD FEED)

**Detecting the error**

The input current on STANDARD FEED is monitored by software. The software will issue an over-current error notification if the current exceeds 30 A.

The software implements a function allowing temporary over-current. This is useful in situations where high peak currents are required.

The battery voltage at the STANDARD FEED terminal is monitored by software. If the voltage is below 5 V for more than 30 seconds, the error message "Battery voltage too low" is triggered. If the voltage exceeds 32 V for more than 64 seconds, the error message "Battery voltage too high" is triggered.

**Root cause of the error**

This error may be caused by the following root causes:

- The maximum current of 30 A has been exceeded.
- The time limit for over-current has been exceeded.
- The battery voltage has dropped below 5 V for more than 30 seconds.
- The battery voltage has exceeded 32 V for more than 64 seconds.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Overcurrent | 0x2323 | 2 |
| Battery voltage is too low | 0x3023 | 2 |
| Battery voltage is too high | 0x3024 | 2 |

**Fixing the root cause**

The controller must respond to the error message and disable the outputs on the module JXM-IO-E02.

## Protected feed power input (PROTECTED FEED)

**Detecting the error**

The input current on PROTECTED FEED is monitored by software. The software will issue an over-current error notification if the current exceeds 30 A.

The solid state switch used to disable the safety outputs (safety switch) also implements a hardware limit. The safety switch will switch off automatically if the switch temperature rises too high and the set actual current is exceeded. The actual current that will cause the safety switch to disconnect is dependent on the ambient temperature.

The software implements a function allowing temporary over-current. This is useful in situations where high peak currents are required.

**Root cause of the error**

This error may be caused by the following root causes:

- The maximum current of 30 A has been exceeded.
- The time limit for over-current has been exceeded.
- The safety switch temperature is too high and the actual current is at least 30 A.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Overcurrent | 0x2322 | 2 |
| Safety Switch Failure | 0x5001 | 8 |

If the switch fails in the ON state, the JXM-IO-E02 will additionally issue the "Safety Switch Failure" notification.

## Digital Outputs 1 ... 8 (Standard Outputs)

**Detecting the error**

A threshold can be programmed for both over-current and cable breakage (no-load) via the System Parameters interface. Over-current limit can be set to between 100 mA and 2.5 A per channel. The no-load threshold can be set between 50mA and 250mA.

Note that this no-load threshold is shared for all digital outputs. No-load detection can be enabled or disabled for individual output channels. A no-load fault can only be detected when a channel is switched on (enabled).

The software implements a function allowing temporary over-current. This is useful in situations where high peak currents are required.

A short to ground is reported in the event of a massive overcurrent.

If an output has experienced at least 4 shorts to ground and/or overcurrent faults, the software will lock this output to prevent it being switched on again.

**Root cause of the error**

This error may be caused by the following root causes:

- The programmed limit for over-current has been exceeded.
- The load current has exceeded 10 A and the over-current situation has exceeded 180 ms.
- The programmed limit for no-load has been exceeded.
- One output has experienced at least 4 shorts to ground and overcurrent faults.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| 1 | The module will send a CANopen® emergency object to the controller. |
| 2 | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Short to GND | 0x9000 - 0x9007 | 1 |
| Overcurrent | 0x2300 - 0x2307 | 1 |
| No-load (cable breakage) | 0x23A0 - 0x23A7 | 1 |
| Lock-out | 0x5300 | 0 |

## Digital Outputs 9 ... 16 (Safety Outputs)

**Detecting the error**

A threshold can be programmed for both over-current and cable breakage (no-load) via the System Parameters interface. Over-current limit can be set to between 100 mA and 5 A per channel. The no-load threshold can be set between 50mA and 250mA.

Note that this no-load threshold is shared for all digital outputs. No-load detection can be enabled or disabled for individual output channels. Therefore, a no-load fault can only be detected when a channel is switched on (enabled).

The software implements a function allowing temporary over-current. This is useful in situations where high peak currents are required.

A short to ground is reported in the event of a massive overcurrent.

If an output has experienced at least 4 shorts to ground and/or overcurrent faults, the software will lock this output to prevent it being switched on again.

**Root cause of the error**

This error may be caused by the following root causes:

- The programmed limit for over-current has been exceeded.
- The load current has exceeded 10 A and the over-current situation has exceeded 180 ms.
- The programmed limit for no-load has been exceeded.
- One output has experienced at least 4 shorts to ground and overcurrent faults.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Short to GND | 0x9010 - 0x9017 | 1 |
| Overcurrent | 0x2310 - 0x2317 | 1 |
| No-load (cable breakage) | 0x23B0 - 0x23B7 | 1 |
| Lock-out | 0x5310 | 0 |

## Analog output

**Detecting the error**

The analog output will detect short circuit to ground faults.

An over-current fault is triggered in software when the measured load current exceeds the maximum output current specified in the system parameters.

A power dissipation fault is triggered when the internal power dissipation exceeds 2 W. Internal dissipation is calculated according to the following formula:

$P_D=(V_{STD\text{-}Feed}-V_{out}) \times I_{out}$

A short to ground is reported in the event of a massive overcurrent.

If an output has experienced at least 5 shorts to ground and/or overcurrent faults, the software will lock this output to prevent it being switched on again.

**Root cause of the error**

This error may be caused by the following root cause:

- When a short to ground is detected, the output is disabled and the fault notification is sent out.
- The programmed load current value of 250 mA has been exceeded.
- One output has experienced at least 5 shorts to ground and overcurrent faults.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error.<br>The analog output will remain disabled until the module is instructed to set the analog output to a normal mode again or until a power cycle has occurred. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Short to GND | 0x9020 | 1 |
| Overcurrent | 0x2370 | |
| Power dissipation | 0x4220 | |
| Lock-out | 0x5500 | |

## PWM outputs 1 ... 3

**Detecting the error**

The PWM outputs can be used in one of two modes:

- Current-controlled output
- PWM output with static duty cycle.

When these outputs are used as current-controlled outputs, the JXM-IO-E02 will detect short circuit to ground and no-load faults. No-load is defined by a current threshold which is user selectable through the System Parameters interface.

When a PWM output is set as a static duty-cycle output, the JXM-IO-E02 will additionally detect over-current faults. These faults are also defined by a user selectable current threshold.

A short to ground is reported in the event of a massive overcurrent.

If an output has experienced at least 5 shorts to ground and/or overcurrent faults, the software will lock this output to prevent it being switched on again.

**Root cause of the error**

This error may be caused by the following root causes:

- The programmed limit for over-current has been exceeded.
- The programmed limit for no-load has been exceeded.
- A short-circuit to ground has occurred.
- One output has experienced at least 5 shorts to ground and overcurrent faults.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Short to GND | 0x90D0 - 0x90D2 | 1 |
| Overcurrent | 0x23D0 - 0x23D2 | 1 |
| No-load (cable breakage) | 0x23C0 - 0x23C2 | 1 |
| Lock-out | 0x5100 | 0 |

## H-bridge

**Detecting the error**

Full protection is only available when the H-Bridge is used in the H-Bridge PWM output modes.

If the H-Bridge is used as two independent digital outputs, only short-circuit to ground fault detection is possible. This feature is unable to detect which of the outputs has caused the problem and will disable all four outputs in the case of a problem.

Overcurrent and no-load faults have user selectable thresholds. These can be set through the System Parameters interface.

Short circuit fault is triggered when a massive overcurrent occurs either in a high-side switch or in a low-side switch. This can trigger regardless of the output mode of the H-bridge.

If an output has experienced at least 5 short-circuit and/or overcurrent faults, the software will lock this output to prevent it being switched on again.

**Root cause of the error**

This error may be caused by the following root causes:

- The programmed limit for over-current has been exceeded.
- The programmed limit for no-load has been exceeded.
- A short-circuit to ground has occurred.
- One output has experienced at least 5 short-circuit and overcurrent faults.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Short to GND | 0x9021 | 1 |
| Overcurrent | 0x2321 | 1 |
| No-load (cable breakage) | 0x2331 | 1 |
| Lock-out | 0x5200 | 0 |

## Switch feed outputs 1 ... 2

**Detecting the error**

Although the fault condition is "Overtemperature", this fault encompasses both, short-circuit to ground and over-current faults. If either fault occurs, the module JXM-IO-E02 will issue an over-temperature error for the output.

A short to ground is reported in the event of a massive overcurrent.

If an output has experienced at least 5 faults, the software will lock this output to prevent it being switched on again.

**Root cause of the error**

This error may be caused by the following root causes:

- The limit for over-current has been exceeded.
- A short-circuit to ground has occurred.
- One output has experienced at least 5 faults.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Overtemperature | 0x4231 - 0x4232 | 4 |
| Lock-out | 0x5400 | 0 |

## Safety switch (solid-state relay)

**Root cause of the error**

This error may be caused by the following root cause:

- The safety switch (solid-state relay) fails to disable the safety outputs.

**Response of the module to this error**

The module will set the corresponding bit in the CANopen® error register and will send the following error code to the controller:

| Error type | Error code | Error register |
|---|---|---|
| Hardware error | 0x5001 | 8 |

## 5 V reference output

**Root cause of the error**

This error may be caused by the following root causes:

- The limit for overcurrent has been exceeded.
- A short-circuit to ground has occurred.
- 5 V output has turned off.

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Overcurrent | 0x2320 | 2 |
| 5 V output error | 0x3020 | 2 |

## Generic fault detection

| | |
|---|---|
| **Fault description** | The JXM-IO-E02 can also detect certain faults which are not directly linked to a specific input or output. These faults are listed in the table below:<br>▪ Internal communication failure<br>▪ Internal temperature<br>▪ System parameter reset<br>▪ RX PDO timeout<br>▪ CAN passive mode<br>▪ Bus off recorder |
| **Internal communication failure** | If the internal communication of the JXM-IO-E02 fails, this error is reported. If this event occurs, certain inputs and/or output may no longer be controllable and the external controller should consider it a serious failure. |
| **Internal temperature failure** | The internal temperature sensor will trigger this fault if the internal temperature exceeds 125 °C. If this fault is triggered, the JXM-IO-E02 must be powered down and allowed to cool down before normal use continues. |
| **Reset of system parameter conflict** | At start-up, the software compares the system parameters stored in two different memories. If there is a conflict, the software will default to the copy kept by the slave controller. This requires an internal restart to reprogram the system parameters. Triggering of this fault is for informational purposes.<br>However, if the system parameter reset fault is triggered continuously, this informs of a more serious failure. Normal operation is not possible at this time. |
| **RX PDO timeout** | When activated in the system parameters, the JXM-IO-E02 will expect to receive the RX PDOs in the specified time. If the RX PDO message does not arrive, the fault is triggered and all outputs controlled by this PDO are disabled. |

**Response of the module to this error**

The module responds to this error in the following levels:

| Step | Description |
|---|---|
| **1** | The module will send a CANopen® emergency object to the controller. |
| **2** | The module will block the function that has caused the error. |

The module will set the corresponding bit in the CANopen® error register and will send the following error code:

| Error type | Error code | Error register |
|---|---|---|
| Internal communication failure | 0x5002 | 5 |
| Internal temperature | 0x4200 | 4 |
| System parameter reset | 0x8001 | 0 |
| RX-PDO-1 timeout | 0x8101 | |
| RX-PDO-2 timeout | 0x8102 | |
| RX-PDO-3 timeout | 0x8103 | |
| RX-PDO-4 timeout | 0x8104 | |
| CAN passive mode | 0x8120 | REC < 127 or TEC < 127 |
| Bus off recorder | 0x8140 | TEC > 255 |

# Appendix

**Introduction**

This appendix contains electrical and mechanical data, as well as operating data.

**Contents**


| Topic | Page |
|---|---|
| Technical data | 128 |
| Index | 139 |

# A: Technical data

**Introduction**

This chapter contains information on electrical and mechanical data, as well as on operating data of the JXM-IO-E02.

**Contents**


| Topic | Page |
|---|---|
| Technical specifications | 129 |
| Physical dimensions | 135 |
| Operating parameters - Environment and mechanics | 137 |
| Operating parameters - EMC | 138 |

# Technical specifications

**Connector**

| Parameter | Description |
|---|---|
| Manufacturer/Model | Tyco AMP |
| Item number | 963484 |
| Design | 70-pin |
| Coding | A 1 |

**Electrical data - Power supply**

| Parameter | Description |
|---|---|
| Operating voltage | DC 8.0 ... 32.0 V |
| Operating voltage - IGNITION FEED | Minimum DC 5.9 V |
| Peak Current: | |
| IGNITION FEED | 2.0 A max. |
| STANDARD FEED | 20.0 A max. (fused up to 30 A) |
| SAFETY FEED | 20.0 A max. (fused up to 30 A) |
| Over-current detection | Yes |

**Communication**

| Parameter | Description |
|---|---|
| Bus type | CAN bus |
| Protocol | CANopen® |
| Baud rate | 250 kBaud (1 MBaud max.) |
| Terminating resistor | Can be activated by means of software |

**Technical data - Tri-state inputs**

| Parameter | Description |
|---|---|
| Purpose | ■ For device coding<br>■ As digital inputs |
| Type of inputs | Pull-up resistor 22 kΩto IGNITION FEED and pull-down resistor 276 Ω to ground |
| Tri-state detection | Tri-state operation is detected by a pull-down resistor to ground. |
| Rated voltage | IGNITION FEED |
| Operating point OFF: | ≤ 1.0 V |
| Operating point ON: | ≥ 4.0 V |

**Technical data - Digital inputs IN 1 ... IN 5**

| Parameter | Description |
|---|---|
| Type of inputs | Software selectable with either 2 kΩ pull-up resistor to IGNITION FEED or 2 kΩ pull-down resistor to ground. |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Operating point OFF: | ≤ 1.0 V |
| Operating point ON: | ≥ 3.5 V |

**Technical data - Digital inputs IN 6 ... IN 13**

| Parameter | Description |
|---|---|
| Type of inputs | Can be configured as active-high inputs |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Operating point OFF: | < 51 % of IGNITION FEED |
| Operating point ON: | > 51 % of IGNITION FEED |
| Input impedance | 100 kΩ |

**Technical data - Digital outputs (STANDARD FEED)**

| Parameter | Description |
|---|---|
| Type of outputs | Active-high output |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Signal voltage OFF | < 1.0 V |
| Signal voltage ON | $U_{STANDARD}$ - 0.5 V |
| Load current of OUT 1 ... OUT 8 | 2.5 A max. |
| Short-circuit capability | Yes |
| Over-current detection | Yes |
| No-load detection | Yes |

**Technical data - Digital inputs IN 14 ... IN 21**

| Parameter | Description |
|---|---|
| Type of inputs | Can be configured as active-high inputs |
| Rated voltage | PROTECTED FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Operating point OFF: | < 51 % of IGNITION FEED |
| Operating point ON: | > 51 % of IGNITION FEED |
| Input impedance | 100 kΩ |

**Technical data - Digital outputs (PROTECTED)**

| Parameter | Description |
|---|---|
| Type of outputs | Active-high output |
| Rated voltage | PROTECTED FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Signal voltage OFF | < 1.0 V |
| Signal voltage ON | $U_{PROTECTED}$ - 0.5 V |
| Load current of PROTECTED OUT 9 ... PROTECTED OUT 16 | 5.0 A max. |
| Can be switched off by electronic safety switch | Yes |
| Short-circuit capability | Yes |
| Over-current detection | Yes |
| No-load detection | Yes |

**Technical data - Switch outputs**

| Parameter | Description |
|---|---|
| Type of switch outputs | Active-high output |
| Possible use | As digital active-low inputs |
| Rated voltage | STANDARD FEED |
| Permissible voltage range | DC 8 ... 32 V |
| Signal voltage OFF | < 1.0 V |
| Signal voltage ON | $U_{STANDARD}$ - 0.5 V |
| Load current | Each 2.5 A max. |
| Short-circuit capability | Yes, thermal (undefined current) |
| Over-current detection | Detection whether the output has switched off. |
| No-load detection | Yes |

**Technical data - PWM outputs**

| Parameter | Description |
|---|---|
| Operating Modes | ■ Current-controlled output<br>■ PWM output with static duty cycle |
| PWM frequency | Configurable values:<br>0 = 1.954 kHz<br>1 = 977 Hz<br>2 = 488 Hz<br>3 = 244 Hz<br>4 = 122 Hz<br>5 = 61 Hz<br>6 = 30 Hz |
| Resolution | 10 bits |
| Load current | 0 ... 2.5 A |
| Short-circuit capability | Yes |
| Over-current detection | Yes |
| No-load detection | Yes |

**Technical data - Analog output**

| Parameter | Description |
|---|---|
| Voltage range at 50 mA | 0 ... STANDARD FEED |
| Current range | 0 ... 100 mA |
| Resolution | 10 bits |
| Electrical isolation | None |
| Short circuit detection | Yes |

**Technical data - Analog inputs**

| Parameter | Description |
|---|---|
| Voltage range | ■ 0 ... 5 V<br>■ 0 ... 40 V |
| Current range | ■ 0 ... 20 mA |
| Input impedance at 0 ... 5 V | 100 kΩ |
| Input impedance at 0 ... IGNITION FEED | 50 kΩ |
| Input impedance at 0 ... 20 mA | 240 Ω |
| Resolution | 10 bits |
| Electrical isolation | None |

### Technical data - Frequency inputs

| Parameter | Description |
|---|---|
| Purpose | ■ As frequency counter<br>■ As digital input |
| Type of inputs | Software selectable with either 2 kΩ pull-up resistor to STANDARD FEED or 2 kΩ pull-down resistor to ground. |
| Frequency measurement range | 5 Hz ... 20 kHz |
| Measurement method | Time-based |
| Result of measurement | Period of the signal in nanoseconds |
| Resolution | 62.5 ns |
| As of hardware revision ≥ 12.04 the following operating point parameters apply: | |
| Operating point OFF: | < 1.7 V for mode 1 ... 4 and IGNITION FEED = 12 V |
| Operating point ON: | > 3.6 V for mode 1 ... 4 and IGNITION FEED = 12 V |
| Operating point OFF: | < 3.4 V for mode 1 ... 4 and IGNITION FEED = 24 V |
| Operating point ON: | > 7.2 V for mode 1 ... 4 and IGNITION FEED = 24 V |
| Operating point OFF: | AC signal < -350 mV for mode 5 |
| Operating point ON: | AC signal > 350 mV for mode 5 |

### Technical data - H-bridge

| Parameter | Description |
|---|---|
| Purpose | ■ As H-bridge<br>■ As two independent digital inputs<br>■ As two independent PWM outputs (active-high and active-low) |
| Rated output current | 2.5 A max. |
| Accuracy of current measurement for H-bridge | < 100 mA |
| Short-circuit capability | Yes |
| Overcurrent detection | Yes |
| No-load detection | Yes |

### Technical data - Regulated output

| Parameter | Description |
|---|---|
| Controlled voltage | DC 5 V |
| Load current | 250 mA max. |
| Over-current detection | Yes |

**Protective and diagnostic functions**

| Malfunction | Response |
|---|---|
| Short circuit | ■ The faulty function is disabled automatically<br>■ A CANopen® emergency object is sent to the controller<br>■ The error message is stored to a history list which is compatible with the CANopen® standard |
| Overload | |
| No-load (cable breakage) | |

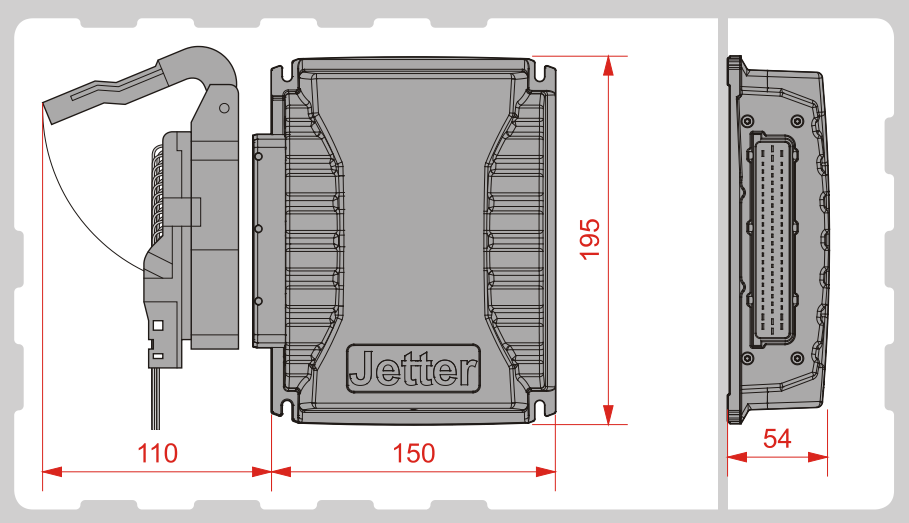
# Physical dimensions

**Introduction**

This chapter details the physical dimensions of the JXM-IO-E02 and the conditions for its installation.

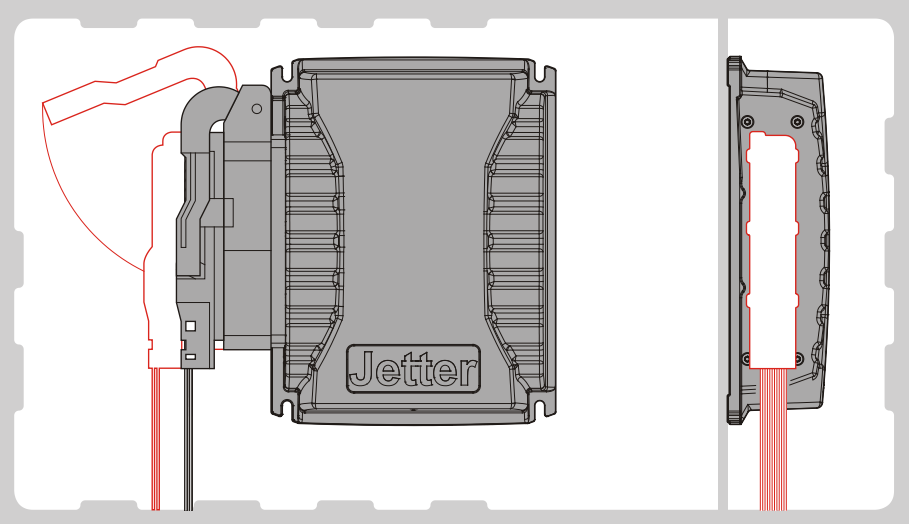
**Physical dimensions**

The illustration shows the physical dimensions of the JXM-IO-E02.




**Space required for installation and service**

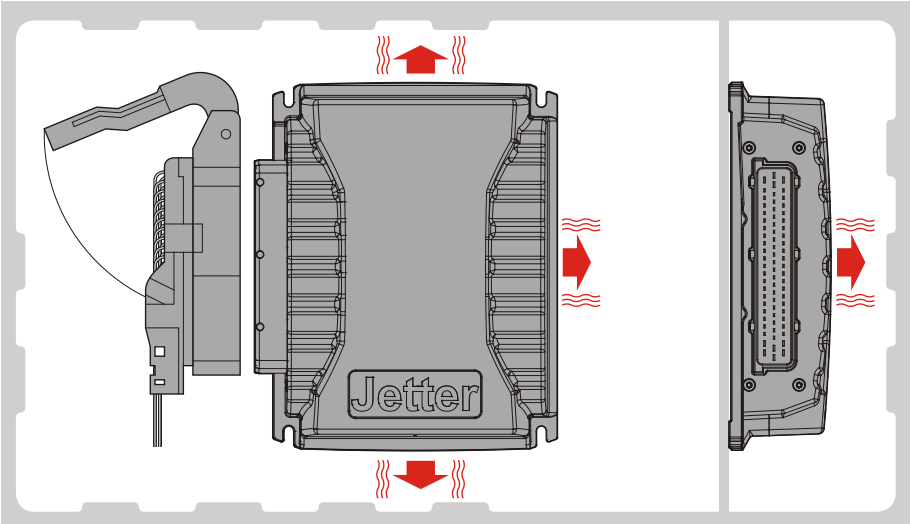
The diagram shows the space required for the JXM-IO-E02.




Ensure there is enough space around the connector for servicing requirements. It should be possible to disconnect the connector at any time.

**Space required to protect against overheating**

The diagram indicates the safety distances to protect against overheating.




Please note:

- The JXM-IO-E02 increases the temperature of the environment as a result of heat emission under load.
- The JXM-IO-E02 operates without interruption at an ambient temperature of up to +85 °C.

Consider the heat emission from the device, in particular when installing it in a critical environment:

- In the vicinity of the fuel tank
- In the vicinity of the fuel pipe
- In the vicinity of flammable vehicle components
- In the vicinity of thermally malleable vehicle components

# Operating parameters - Environment and mechanics

## Environment

| Parameter | Value | Standard |
|---|---|---|
| Operating temperature range | -40 ... +85 °C | |
| Storage temperature range | -40 ... +85 °C | DIN EN 61131-2<br>DIN EN 60068-2-1<br>DIN EN 60068-2-2 |
| Air humidity | 10 ... 95 % | DIN EN 61131-2 |
| Climate test | Humid heat | DIN EN 60068-2-30 |
| Pollution degree | 2 | DIN EN 61131-2 |

## Mechanical parameters

| Parameter | Value | Standard |
|---|---|---|
| Vibration resistance | Vibration, broadband noise | DIN EN 60068-2-6<br>Severity level 2 |
| Shock resistance | 30 g occasionally, 18 ms, sinusoidal half-wave, 3 shocks in the directions of all three spatial axes | DIN EN 60068-2-27 |
| Degree of protection | IP68 only if the individual wires of the mating connector are sealed | DIN EN 60529 |

## Operating parameters - EMC

**EMC - Emitted interference**

This device has been tested as per Directive 72/245/EEC with all amendments up to 2009/19/EC and meets all test criteria.

**EMC - Immunity to interference**

| Parameter | Value | Standard |
|---|---|---|
| Interference immunity to conducted faults | Compliant | Directive 72/245/EEC with all amendments including 2009/19/EC |
| Interference immunity to external magnetic field | 20 ... 1,000 MHz: 100 V/m 1,000 ... 2,000 MHz MHz: 30 V/m | Directive 72/245/EEC with all amendments including 2009/19/EC |
| Load dump | Impulse 5b 70 V | ISO 7637-2 |

# B: Index


## C

CANopen® bus cable - Specification • 29
CANopen® objects • 57
  Digital inputs and outputs • 62, 64
  Setting/reading the node ID • 35, 66
  System Parameters • 86, 98
Components of the JXM-IO-E02 • 15
Connector specification • 27

## D

Diagnostic Features • 113
Disposal • 10

## E

EDS • 23, 85
EEPROM • 106
Error description • 113
Error handling • 113

## I

Identification
  by means of the nameplate • 21
  Setting/reading the node ID • 35, 66
  via CANopen® bus • 22
Initial commissioning • 51
Installation • 46
Intended conditions of use • 10
Interfaces
  Analog I/Os • 42
  CAN • 35
  Digital inputs and outputs • 37
  Frequency inputs • 42
  H-bridge • 42
  Power supply • 33
  Regulated output (5 volts) • 33
  Switch feed outputs • 37
  Wiring - Example • 31

## M

Maintenance • 10
Modifications • 10

## O

Operating parameters
  EMC • 138
  Environment and mechanics • 137
Operating system update • 85
Order reference - JXM-IO-E02 • 16
OS update • 85

## P

PDO Specification • 108
Personnel qualification • 10
Physical dimensions • 17
Product description – JXM-IO-E02 • 14
Protective features • 113

## R

Repair • 10

## S

Safety instructions • 9
Setting/reading the node ID • 35, 66
Software version • 23, 102
System Parameters • 86, 98

## T

Technical specifications • 129
Transport • 10
Nameplate • 21

## U

Usage other than intended • 10

Jetter automation

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg | Germany

Phone +49 7141 2550-0
Fax +49 7141 2550-425
info@jetter.de
www.jetter.de

We automate your success.